



# NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

## THESIS

**REENGINEERING OF THE DEFENSE BIOMETRIC  
IDENTIFICATION SYSTEM (DBIDS) EQUIPMENT  
TRACKING DATABASE**

by

W. Tracy Young  
Dalton H. Clarke

September 2009

Thesis Advisor:  
Second Reader:

Karl D. Pfeiffer  
Albert Barreto

**Approved for public release; distribution is unlimited**

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
<p>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.</p>			
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED	
	September 2009	Master's Thesis	
4. TITLE AND SUBTITLE		5. FUNDING NUMBERS	
Reengineering of the Defense Biometric Identification System (DBIDS) Equipment Tracking Database			
6. AUTHOR(S)		7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	
W. Tracy Young and Dalton H. Clarke		Naval Postgraduate School Monterey, CA 93943-5000	
8. PERFORMING ORGANIZATION REPORT NUMBER		9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)	
		N/A	
10. SPONSORING/MONITORING AGENCY REPORT NUMBER		11. SUPPLEMENTARY NOTES	
		The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.	
12a. DISTRIBUTION / AVAILABILITY STATEMENT		12b. DISTRIBUTION CODE	
Approved for public release; distribution is unlimited			
13. ABSTRACT (maximum 200 words)			
<p>The Defense Manpower Data Center (DMDC) manages the Defense Biometric Identification System (DBIDS). DBIDS captures personal and biometric information to manage DoD-wide access, control, and personnel accountability. DBIDS equipment is installed globally, and managed by a central office on the Monterey peninsula. Program managers track data about the numbers and type of equipment installed at each site. Program managers were tracking DBIDS data using a single Microsoft Excel workbook comprised of several, interlinking worksheets (DBIDS Master Plan Spreadsheet). Data updates were error-prone and difficult, requiring close coordination to keep the number of "current" versions of the spreadsheet to a minimum. This thesis initially focused on reviewing the business rules and processes surrounding DBIDS document, and then transitioned into designing, developing, and implementing of a relational database solution to improve problem areas identified during the initial review. After implementation of the database, this thesis explored the effects of making such a change within an organization. This was attempted by identifying and measuring changes in performance and accuracy of the system; by measuring pre- and post-user satisfaction through the qualitative methods of questionnaires and interviews; and finally using this analysis to improve the project through maintenance and growth iterations.</p>			
14. SUBJECT TERMS			15. NUMBER OF PAGES
Database, Database Development Life Cycle, Rapid Prototyping, Business Process Management Software, Business Process Improvement, Business Process Redesign			211
16. PRICE CODE			
17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release; distribution is unlimited**

**REENGINEERING OF THE DEFENSE BIOMETRIC IDENTIFICATION SYSTEM  
(DBIDS) EQUIPMENT TRACKING DATABASE**

W. Tracy Young  
Captain, United States Marines  
B.S., University of Utah, 2003

Submitted in partial fulfillment of the  
requirements for the degree of

**MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT**

from the

**NAVAL POSTGRADUATE SCHOOL  
September 2009**

Dalton H. Clarke  
Lieutenant, United States Navy  
B.S., Norfolk State University, 2003

Submitted in partial fulfillment of the  
requirements for the degree of

**MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT**

from the

**NAVAL POSTGRADUATE SCHOOL  
March 2009**

Authors: W. Tracy Young

Dalton H. Clarke

Approved by: Karl D. Pfeiffer  
Thesis Advisor

Albert Barreto  
Second Reader

Dan Boger  
Chairman, Information Sciences Department

THIS PAGE INTENTIONALLY LEFT BLANK

## ABSTRACT

The Defense Manpower Data Center (DMDC) manages the Defense Biometric Identification System (DBIDS). DBIDS captures personal and biometric information to manage DoD-wide access, control, and personnel accountability. DBIDS equipment is installed globally, and managed by a central office on the Monterey peninsula. Program managers track data about the numbers and type of equipment installed at each site. Program managers were tracking DBIDS data using a single Microsoft Excel workbook comprised of several, interlinking worksheets (DBIDS Master Plan Spreadsheet). Data updates were error-prone and difficult, requiring close coordination to keep the number of “current” versions of the spreadsheet to a minimum. This thesis initially focused on reviewing the business rules and processes surrounding DBIDS document, and then transitioned into designing, developing, and implementing of a relational database solution to improve problem areas identified during the initial review. After implementation of the database, this thesis explored the effects of making such a change within an organization. This was attempted by identifying and measuring changes in performance and accuracy of the system; by measuring pre- and post-user satisfaction through the qualitative methods of questionnaires and interviews; and finally using this analysis to improve the project through maintenance and growth iterations.

THIS PAGE INTENTIONALLY LEFT BLANK

## TABLE OF CONTENTS

I.	INTRODUCTION.....	1
A.	BACKGROUND .....	1
B.	PURPOSE.....	2
C.	STATEMENT OF PROBLEM.....	2
D.	ASSUMPTIONS .....	3
1.	Resource Assumption.....	3
2.	Benefit Assumptions .....	3
3.	Environmental Assumptions .....	3
4.	Budgetary Assumptions .....	4
E.	SCOPE AND METHODOLOGY.....	4
F.	ORGANIZATION.....	4
II.	CURRENT DOMAIN OVERVIEW.....	7
A.	DBIDS MASTERPLAN SPREADSHEET OPERATIONAL ENVIRONMENT .....	7
1.	As-Is System Configuration.....	8
2.	Users of the As-Is System Roles and Responsibilities .....	9
B.	AS-IS PROCESS DESCRIPTION .....	9
C.	SYSTEM ANALYSIS.....	11
1.	Problem Analysis.....	11
a.	<i>Cause and Effect</i> .....	11
D.	SUMMARY .....	13
III.	METHODOLOGY FOR PROCESS IMPROVEMENT .....	15
A.	BUSINESS PROCESS IMPROVEMENT .....	15
1.	Business Process Improvement Overview.....	15
a.	<i>FAST</i> .....	16
b.	<i>Benchmarking</i> .....	16
c.	<i>Business Process Reengineering</i> .....	17
d.	<i>Process Redesign</i> .....	17
B.	MODELING AS-IS BASELINE PROCESS .....	19
1.	As Is Process Description.....	20
a.	<i>Initial Configuration of Savvion</i> .....	20
2.	As-Is Process Improvement Options .....	21
3.	To-Be Process Description .....	22
a.	<i>To-Be Initial Configuring of Savvion</i> .....	22
4.	Comparison of the As-Is and To-Be Systems .....	23
C.	SUMMARY .....	24
IV.	DBIDS DATABASE DESIGN & DEVELOPMENT.....	25
A.	DBIDS DATABASE DESIGN.....	25
B.	DEVELOPMENT LIFE CYCLE .....	25
C.	PLANNING.....	26
1.	Scope .....	26

2.	<b>Key Issues .....</b>	26
3.	<b>Assumptions .....</b>	26
4.	<b>Expectations .....</b>	27
5.	<b>Data Requirements .....</b>	27
6.	<b>Tentative Schedule .....</b>	27
<b>D.</b>	<b>FEASIBILITY .....</b>	28
1.	<b>Technological Infrastructure .....</b>	28
2.	<b>Operational Skills .....</b>	28
3.	<b>Economics.....</b>	29
<b>E.</b>	<b>REQUIREMENTS DEFINITION .....</b>	29
1.	<b>Detailed Review of the As-Is Spreadsheet.....</b>	29
2.	<b>Identified Requirements for the Project.....</b>	30
<b>F.</b>	<b>DESIGN.....</b>	31
1.	<b>Logical Design .....</b>	31
2.	<b>Physical Design .....</b>	34
3.	<b>Graphical User Interface (GUI) Design.....</b>	36
4.	<b>Prototypes .....</b>	37
	<i>a. Version 0. –16 October 2008 .....</i>	37
	<i>b. Version 1.0–4 December 2008.....</i>	37
	<i>c. Version 1.11–9 January 2009 .....</i>	38
	<i>d. Version 1.20–1 March 2009 .....</i>	38
	<i>e. Version 1.21–13 March 2009 .....</i>	38
5.	<b>Design Phase Results .....</b>	38
	<i>a. Key Based Data Model.....</i>	39
	<i>b. Entity Definition Matrix.....</i>	40
	<i>c. Stake Holder Glossary.....</i>	40
	<i>d. Use Case Glossary.....</i>	40
	<i>e. Social Network .....</i>	40
	<i>f. Switchboard Mapping.....</i>	40
<b>G.</b>	<b>DEVELOPMENT .....</b>	41
1.	<b>DBMS .....</b>	41
	<i>a. Physical Design Created in Access .....</i>	41
	<i>b. Data Conversion .....</i>	41
2.	<b>GUI Development.....</b>	42
	<i>a. Structured Query Language (SQL).....</i>	42
	<i>b. Visual Basic (VB) Coding .....</i>	43
	<i>c. Queries.....</i>	43
	<i>d. Forms .....</i>	44
	<i>e. Reports .....</i>	44
	<i>f. Security.....</i>	45
	<i>g. Auditing .....</i>	46
	<i>h. Menus &amp; Switchboard.....</i>	47
3.	<b>Testing .....</b>	47
<b>H.</b>	<b>SUMMARY .....</b>	47
<b>V.</b>	<b>DBIDS DATABASE IMPLEMENTATION .....</b>	49

A.	<b>IMPLEMENTATION .....</b>	49
1.	<i>Version 1.22–3 April 2009 .....</i>	49
B.	<b>MAINTENANCE &amp; GROWTH .....</b>	49
1.	<i>Initial Maintenance and Growth.....</i>	49
a.	<i>Versions 1.2–1.32–5 April–3 August 2009.....</i>	50
b.	<i>Version 1.33–7 August 2009.....</i>	50
2.	<i>Regular Maintenance Requirements .....</i>	50
3.	<i>Database Management Processes .....</i>	50
4.	<i>Growth .....</i>	51
C.	<b>DOCUMENTATION.....</b>	52
D.	<b>RESULTS.....</b>	52
1.	<i>Requirements.....</i>	53
2.	<i>Customer Satisfaction.....</i>	56
E.	<b>SUMMARY .....</b>	57
VI.	<b>SUMMARY, CONCLUSION AND FUTURE RESEARCH .....</b>	59
A.	<b>SUMMARY .....</b>	59
B.	<b>CONCLUSION .....</b>	59
C.	<b>FUTURE DEVELOPMENT OR RESEARCH .....</b>	61
	<b>LIST OF REFERENCES.....</b>	63
	<b>APPENDIX A—QUESTIONNAIRE RESULTS.....</b>	67
	<b>APPENDIX B—AS-IS INTERVIEWS.....</b>	71
	<b>APPENDIX C—TO-BE INTERVIEWS.....</b>	77
	<b>APPENDIX D—PIECES FRAMEWORK CHECKLIST.....</b>	81
	<b>APPENDIX E—PROCESS SIMULATIONS .....</b>	83
	<b>APPENDIX F—STAKE HOLDER GLOSSARY.....</b>	89
	<b>APPENDIX G—USE CASE GLOSSARY .....</b>	91
	<b>APPENDIX H—SOCIAL NETWORK .....</b>	93
	<b>APPENDIX I—DBIDS MASTER DATABASE HELP FILE.....</b>	95
	<b>APPENDIX J—VISUAL BASIC MODULE .....</b>	187
	<b>INITIAL DISTRIBUTION LIST .....</b>	191

THIS PAGE INTENTIONALLY LEFT BLANK

## LIST OF FIGURES

Figure 1.	(Defense Biological Identification System (DBIDS), 2006) .....	8
Figure 2.	General As-Is Process.....	10
Figure 3.	Redesign Process (El Sway, 2001) .....	19
Figure 4.	Entity Relationship Model .....	32
Figure 5.	Access Key-based Relational Model .....	36
Figure 6.	Key Based Data Model.....	39
Figure 7.	Audit Trail View Form .....	101
Figure 8.	Audit Trail Report.....	102
Figure 9.	Compacting & Repairing.....	102
Figure 10.	Database Manager Window .....	103
Figure 11.	Exit Message.....	104
Figure 12.	Open Exclusive .....	104
Figure 13.	Switchboard Items.....	105
Figure 14.	User and Group Accounts .....	105
Figure 15.	Add New User .....	106
Figure 16.	Adding Group Memberships.....	106
Figure 17.	New Record Button .....	107
Figure 18.	Adding a new user to PM Filters.....	107
Figure 19.	Air Force NORTHCOM PM Filters.....	108
Figure 20.	Change Logon Password .....	108
Figure 21.	Shortcut.....	109
Figure 22.	Access Key-Based Relationship Diagram .....	113

THIS PAGE INTENTIONALLY LEFT BLANK

## LIST OF TABLES

Table 1.	PIECES Matrix .....	12
Table 2.	Measurement Parameters .....	20
Table 3.	As-Is Model Measures.....	24
Table 4.	To-Be Model Measures .....	24
Table 5.	Entity Definition Matrix.....	40

THIS PAGE INTENTIONALLY LEFT BLANK

## LIST OF SYMBOLS, ACRONYMS, AND ABBREVIATIONS

AFRICOM	African Command
Billet	A personnel position or assignment that may be filled by one person.
BPI	Business Process Improvement
BPM	Business Process Management
BPR	Business Process Reengineering
CENTCOM	Central Command
COBRA	Consultants and Opportunities in Business Restructuring & Analysis
COCOM	Combatant Commander
CONUS	Continental United States
DBIDS	Defense Biometric Identification Data System
DBMS	Database Management System
DDLC	The Database Development Life Cycle
Direct Cutover	A method of system change over, when triggered, implements a new system and replaces a legacy system.
DMDC	Defense Manpower Data Center
DoD	Department of Defense
EUCOM	European Command
FAST	Fast Analysis Solution Technique
Foreign Key	A referential constraint linking one entity to another entity's primary key.
GUI	Graphical User Interface
IS	Information System
IT	Information Technology
ITM	Information Technology Management
Meta Data	Data about Data
NORTHCOM	Northern Command
LAN	Local Area Network
Normalize	A systematic way of ensuring that a database structure is suitable for general-purpose querying and free of certain undesirable characteristics (insertion, update, and deletion anomalies) that could lead to a loss of data integrity.
NPS	Naval Postgraduate School

OCONUS	Outside Continental United States
PACOM	Pacific Command
Phase-In Installation	A method of system change over, where incremental changes are made replacing a legacy system with a new system.
PIECES	Performance, Information, Economy, Control Efficiency, and Service
PIT	Process Improvement Team
PM	Project Manager
Primary Key	A property that uniquely identifies an entity.
Relational Database	A database that groups data using common attributes.
SQL	Structured Query Language
Survey Monkey	A tool to create and publish custom surveys to view aggregated results graphically.
VB	Visual Basic
View	The resulting set of data that a query defines.
VPN	Virtual Private Network
WAN	Wide Area Network

## **ACKNOWLEDGMENTS**

We would like to give our sincere thanks to our thesis advisors Dr. Karl Pfeiffer and Professor Albert Barreto for their guidance throughout this project and allowing us the leeway to conduct the thesis research our way. I would also like to thank all the project managers at DMDC who were involved for their support, particularly Michael Milos, who proved to be invaluable during the thesis maturity.

We would like to thank our wives for their continuous support, motivation and their constant reminders to keep working on our thesis. They have been our foundation throughout this thesis process.

THIS PAGE INTENTIONALLY LEFT BLANK

## I. INTRODUCTION

With today's turbulent economy, more emphasis has been placed on increasing governmental oversight, to avoid wasteful spending on Information Technology (IT) projects. IT projects are now subjected to more stringent assessment procedures to promote greater transparency within the DoD. Subsequently, screening of these potential IT projects is conducted by Project Managers (PMs), who are knowledgeable about the standard methodology of IT implementation, but lack experience in Business Process Improvement (BPI) methodologies. PMs measure an IT project's success based upon whether the intended Information System (IS) is delivered "on time" and "within budget". Arguably, many IT organizations are not proactive but reactive in nature. Reactive organizations try to mitigate challenges with business processes by mere IS automation. Approximately 65% to 80% of these IT projects fail to meet their intended objectives. For long-term success in any IT project, the focus should be on combining both BPI and IT automation processes (Weerakkody & Currie, 2003).

In this research, we document the successful design and implementation of a small project using both business improvement methodologies and current IT. The Defense Manpower Data Center (DMDC), a DoD administrative organization, requested this research to assist program managers who own the equipment management sub-process of the Defense Biometric Identification Data System (DBIDS). The software, and associated database developed in this project, has been implemented at DMDC offices throughout the world.

### A. BACKGROUND

PMs, working at DMDC, foresaw that the current DBIDS Master Plan Spreadsheet System used for tracking DBIDS was flawed. In the summer of 2008, the PMs decided to enlist the help of Naval Postgraduate School (NPS)

students currently enrolled in a Database Management course. The PMs required assistance in replacing the DBIDS Master Plan Excel Spreadsheet System with a more mature database management system (DBMS).

This thesis will explore the effects of making such a change within an organization; identify and measure changes in performance and accuracy of the system; and measure As-Is and To-Be user satisfaction using the qualitative methods of questionnaires and interviews. The overall goal of this analysis is to improve the DMDC business processes with respect to DBIDS.

Additionally, this thesis will document the process of designing, developing, transitioning to, and maintenance of a relational database solution to enhance organization operations. The development platform was targeted at Microsoft Access 2003, accessed through an onsite server connected to regional offices around the globe through a wide area network (WAN).

## **B. PURPOSE**

By transitioning from a legacy system with weak scalability and no real network access to a mature DBMS tool, the DMDC office would improve efficiency, effectiveness, along with increased knowledge of the business metrics at no cost. In addition, this study would offer the users some insight into both current and future requirements. Finally, this thesis will provide a real-world example of the development, management, and results of a DBMS using the processes currently taught in NPS's Information Technology Management (ITM) curriculum.

## **C. STATEMENT OF PROBLEM**

At the initial meeting with PMs from DMDC, the project team was presented with the perceived limiting factors of the As-Is system. With the As-Is system, PMs tracked the data pertaining to sites globally using a Microsoft Office 2003 Excel spreadsheet, in a single workbook comprised of several inter-linking worksheets. Data updates were error-prone and difficult, and required close

coordination to keep the number of “current” versions to a minimum. As a result, these limiting factors created a more constrained business process for providing on-time reporting.

## **D. ASSUMPTIONS**

### **1. Resource Assumption**

- The information presented to the project team by the PMs concerning the As-Is processes and architecture were accurately identified.
- The project team would have access to PMs to conduct interviews and questionnaires.
- The As-Is meta data would be readily available.

### **2. Benefit Assumptions**

- The use of BPI methodologies would improve the efficiency and effectiveness, while lowering time costs.
- Knowledge of the business processes would provide greater insight into the organization.
- The implementation of a DBMS would improve the accuracy of data tracking.
- Security features would allow non-repudiation through auditing in the To-Be system.
- The functionality increased by allowing for greater stakeholder satisfaction.

### **3. Environmental Assumptions**

- The To-Be model would comply with DMDC’s IT standards and regulations.
- The DBMS will be hosted on computers with a licensed Microsoft Access 2003 installation.

- Any personal information in the database would comply with current Privacy Laws.

#### **4. Budgetary Assumptions**

- No additional costs will be incurred throughout the duration of the project.

### **E. SCOPE AND METHODOLOGY**

The scope would be limited to development and implementation of a fully functional DBMS delivered to DMDC to aid in the management of the DBIDS program. A detailed analysis of the existing processes expectations, perceptions, and users satisfaction of the completed DBMS.

The methods and resources used would include system analysis, DBMS development, modeling, and simulation software tools, articles, publications, and related material. These references would help establish a baseline for conducting rigorous system analysis and design methods. Interviewing key personnel, conducting limited questionnaires, and initiating a BPI approach will help establish the baseline of the As-Is model. With this baseline established, a listing of requirements for an IT solution can be compiled. Using these requirements, a DBMS and its associated Graphical User Interface (GUI) could be designed, developed, and implemented. Final interviews and questionnaires were conducted once the DBMS system had been implemented, in order to measure the delta in functionality, scalability, effectiveness, efficiency, and cost savings.

### **F. ORGANIZATION**

The rest of this thesis is organized in the following manner:

- Chapter II will provide domain background on DMDC, their current business process and assess the current system.
- Chapter III will introduce the BPI methodology for assessing As-Is model measures.

- Chapter IV will discuss the project team's use of the Database Development Life Cycle up to the completion of the development phase. In parallel to the initiation of this cycle, the simultaneous GUI design will be discussed.
- Chapter V will continue the discussion of the Database Development Life Cycle, with the implementation, maintenance, and growth phases. Additionally, the resulting project and its adherence to scope, expectations, and requirements are discussed.
- Chapter VI will solidify conclusions and make recommendations about where future research or development can expand on these efforts.

THIS PAGE INTENTIONALLY LEFT BLANK

## II. CURRENT DOMAIN OVERVIEW

### A. DBIDS      MASTERPLAN      SPREADSHEET      OPERATIONAL ENVIRONMENT

Implementing IT without first obtaining an explicit understanding of the business environment may only lead to short term results. To understand the challenges to the existing system, it is necessary to first review the organizational structure of the Defense Manpower Data Center (DMDC).

DMDC maintains one of the largest data archives of personnel, training, manpower, and financial data. Furthermore, a broad spectrum of information pertaining to all military service members (Active Duty, National Guard, Reservists, and Retirees) has been collected since 1968 (Department of Defense). Today, with the numerous advances in IT, DMDC utilizes an array of data collection techniques to capture and archive personnel data. The Defense Biometric Identification System (DBIDS) platform (Figure 1) has been fielded globally as a force protection measure. This system captures personnel data and archives it within the DMDC enterprise database.

The purpose of DBIDS is to manage personnel, property, and installation access to DoD facilities through the use of both barcode technology and biometric finger print recognition. The captured information enables quick queries for verification and validation via an Oracle database located at DMDC. The DBIDS system is one in a series of personnel data repositories that are able to share information. Easy access to this information allows law enforcement personnel to quickly collaborate and enhance security response time. This offers tremendous benefits over traditional methods of a security officer checking IDs at the gate.

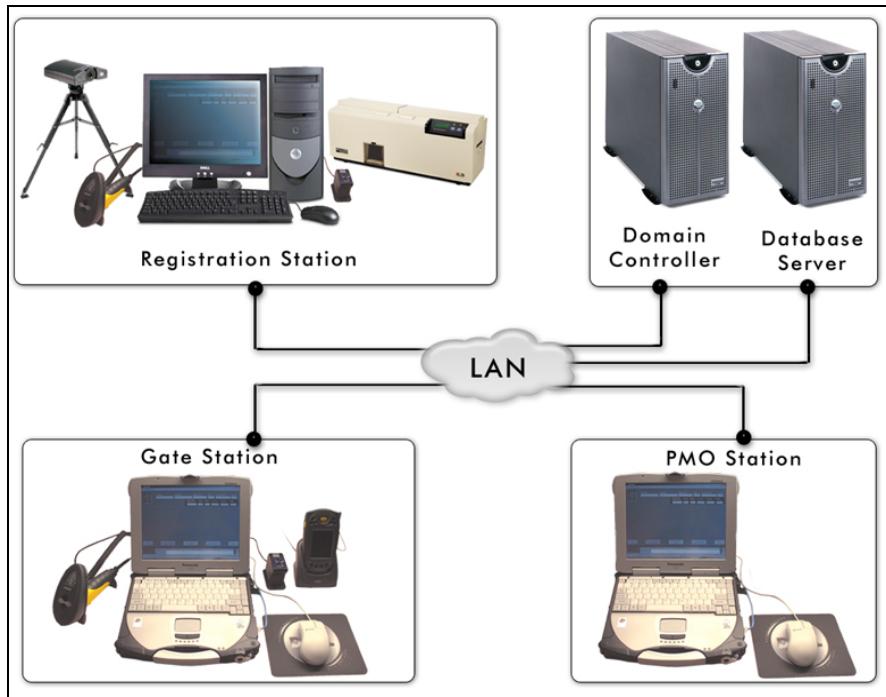


Figure 1. (Defense Biological Identification System (DBIDS), 2006)

Despite the tremendous benefits gained through the fielding of the DBIDS platforms, these systems were not always efficiently or effectively managed. According to one interview in the study (Appendix B—As-Is Interviews), top managers were not able to accurately account for both scheduled and completed DBIDS installations. DMDC attempted to mitigate this oversight, through the implementation of a Microsoft Excel 2003 Spreadsheet. The system was called DBIDS Master Plan Spreadsheet.

### 1. As-Is System Configuration

Throughout the rest of this thesis the DBIDS Master Plan Spreadsheet will be referred to as the As-Is system. The As-Is system was on each PM's desktop and/or mobile computer station. Furthermore, the computer ran a Windows XP Operating System and the Microsoft Office Excel 2003 application. Operationally, the system user would gain access to the As-Is system via the Local Area Network (LAN), WAN or through a Cisco VPN internet connection. Through this access, each user was mapped to an unrestricted shared drive.

Each of the Regional PMs maintain their own copies of the As-Is (Excel spreadsheet). The Operational PMs maintained the master consolidated copy within the share drive.

## **2. Users of the As-Is System Roles and Responsibilities**

- **Managers** were responsible overall for both the effective management of fielded DBIDS platforms and evaluations of Regional PM performance through the As-Is system. Additionally, managers provide strategic guidance to the system users
- **Regional PM (CONUS and OCONUS)** were responsible, according to business rules, for biweekly inputs to the As-Is system. These inputs were to keep the data accurate and update to date. The accuracy and timeliness of these inputs further served as performance measures for evaluation purposes.
- **Operations PM** served as an administrator and consolidator of the As-Is system data. They were responsible for providing maintenance, continued functionality, and performance updates to the As-Is system.

## **B. AS-IS PROCESS DESCRIPTION**

In addition to understanding the history and organizational structure of the As-Is system, it was also necessary that we define the current business processes. Historically, administrative processes were not clearly defined; unlike the processes pertaining to manufacturing operations. In most manufacturing operational processes, managers had tangible items that could be monitored and changed if necessary. In retrospect, having a clearly defined As-Is process helped us to discover where the challenges were located.

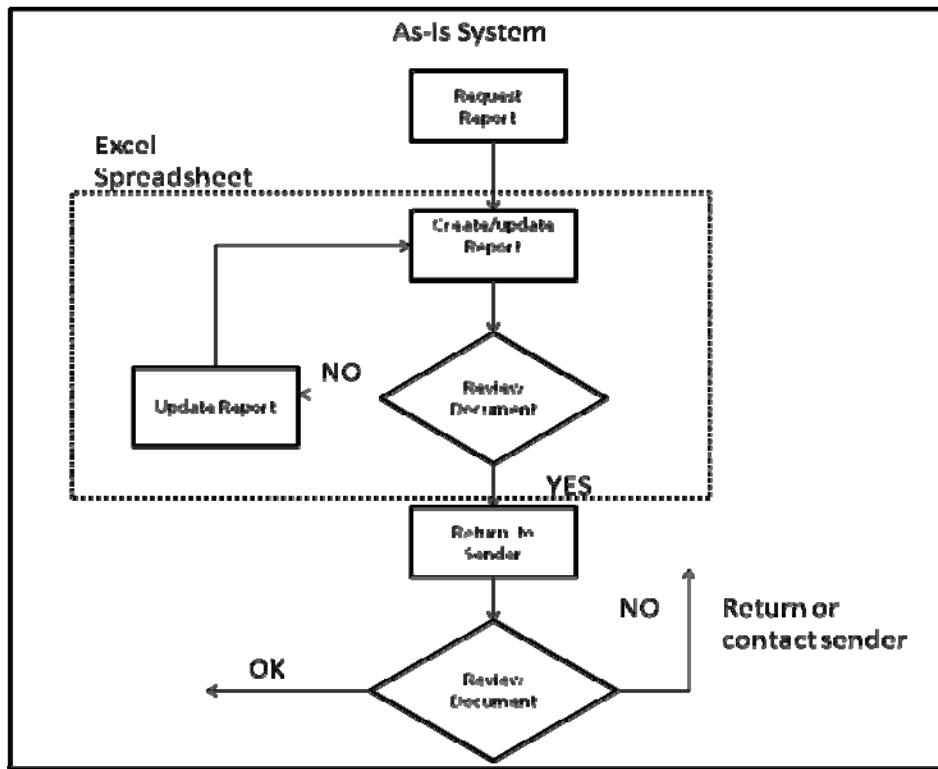


Figure 2. General As-Is Process

The most used As-Is process (Figure 2) will be the focus of this study. The DMDC Stakeholders considered requests for status reports one of the core processes.

- The process began with the Manager requesting a status report on any of the DBIDS fielded. The request was sent to the Operations PM.
- The Operations PM accessed the local As-Is spreadsheet to gather information pertaining to the requested status report.
- The Operations PM reviewed whether the information collected was accurately updated.
  - Inaccurate information was sent for re-working, and regional PMs would update their As-Is spreadsheet. Once updated, the information was sent back to the Operations PM.
  - Accurate information was sent to the Manager for processing.
- The Manager made a final review of the report for any inaccuracies.

## C. SYSTEM ANALYSIS

Previously, PMs from DMDC briefed the Database class at NPS on the scope of their problem. However, the brief excluded details about information pertaining to the architecture and the core process. This study required that the analyst and the system designer develop a clearer understanding of the problem. Questionnaires (Appendix A—Questionnaire Results) were issued in addition to interviews with DMDC Stakeholders. Ten open ended and fixed questions were issued to users of the system through the Survey Monkey Website. The goal of this was to acquire a snapshot of a group of user's attitudes and opinions about the As-Is system. Upon the completion of the questionnaire, the data was aggregated, and a percentage of occurrences were measured.

As previously stated, the goal of the As-Is system was intended as a feedback tool that provided upper management with the status of regional DBIDS projects. Additionally, it served as a performance evaluator for the Regional PMs. Questionnaires, interviews, and facilitated meetings, led to the determination that the system was not well-suited for these objectives.

### 1. Problem Analysis

Conducting the problem analysis can be difficult for novice analysts who look for quick solutions, resulting in misdiagnosing a problem. Misdiagnosing the problem can only complicate matters for the IT project. One example of this, would be a doctor prescribing a solution to his patient without properly assessing him. Applying cause and effect methods will allow for a more in-depth understanding of the problem.

#### a. *Cause and Effect*

The PIECES (Performance, Information, Economy, Control Efficiency, and Service) framework (Wetherbe, 1988) has enabled a number of IT project implementers to quickly categorize and prioritize problems. The PIECES

Framework (Appendix D—PIECES Framework Checklist) used with previous fact-finding methods served as a useful tool to classify problems into four categories (Table 1).

Suggested Problem	P	I	E	C	E	S
Management reports are often not received on time	x	x				
No clear picture of the business metric		x				
Often rework reports	x				x	
Reports were often inaccurate		x		x	x	x
<b>Total</b>	2	3		1	2	1

Table 1. PIECES Matrix

- The system was not efficient with the process of producing reports. According to Operations PMs, they spent most of their time correcting errors that plagued the As-Is system. Often the reports had multiple versions that were being corrected.
- The As-Is system was not effective, because management had no clear picture about the status of any DBIDS system. In addition, there was no way to account for job performance. The As-Is system could not indicate what time or which PM had entered any information into the As-Is spreadsheet. Similar results were revealed in the initial questionnaire, of which 83% of the users agreed that the As-Is system was neither effective nor efficient.
- The As-Is system was plagued with a lack of accuracy in the data. The As-Is spreadsheet had grown to such a size that it was difficult to review site data accurately. The As-Is spreadsheet had over 100 columns of information with possible duplicate entries of sites on over 1000 rows spread across nine different tabs.
- The system lacked any security of the data, allowing anyone to make changes to the data, without raising an alarm since the amount of DBIDS gear being tracked had grown so much.

As depicted in Table 1, while not a precise problem-solving tool, the PIECES framework helped prioritize the problems that should be tackled first. The PIECES matrix indicated that the focus should be on improving Information.

#### **D. SUMMARY**

This Chapter has covered several aspects that are essential to any initiated IT project. The project team first had to define the operational environment in which the As-Is system interacted. This established a general understanding about the purpose of the As-Is system. Having a general understanding of the As-Is system, lends itself to implementing a rigorous system analysis followed by problem analysis. With interviews and questionnaires conducted, the project team then identified potential external problems that existed.

Most importantly, in the subsequent Chapter, this study takes the appropriate step to identify a proper BPI approach, in order to leverage a segregate IT system. These BPI approaches employ measures that are achieved through an explicit design and analysis of the As-Is process.

THIS PAGE INTENTIONALLY LEFT BLANK

### **III. METHODOLOGY FOR PROCESS IMPROVEMENT**

#### **A. BUSINESS PROCESS IMPROVEMENT**

In Chapter II, the study of the As-Is system was focused on defining the problems and their potential causes which existed within the system. Despite defining most of these causes through interviews and questionnaires, the business process had yet to be fully evaluated. The reason was that these interviews lacked the ability to evaluate the problem internally. One example is a patient who knows his ailment, but cannot indicate the exact cause of the ailment. This situation is similar to the dilemma faced by the project team.

The focus of this Chapter is to conduct further analysis of the business processes using specific BPI methodology. However, due to the number of studies conducted pertaining to the methodology on how to correctly initiate BPI, the project team first briefly reviewed and introduced the relevant BPI approaches. Note that this study does not evaluate any organizational bureaucracy issues in implementing BPI methodologies.

##### **1. Business Process Improvement Overview**

The BPI methodology focuses on incremental improvements to the administrative processes using a culmination of tools such as Fast Analysis Solution Technique (FAST), Process Benchmarking, Business Process Redesign, and Process Reengineering. These approaches are classified as conceptual tools within the analyst's tool bag.

At the end of the BPI documented in this research, project improvement measures were taken to expose the areas of performance constraints. The goal of the BPI was to reduce the cycle time, reduce the cost, and lowering the amount of errors. The assumption was that all of the above improvements could be met.

**a. *FAST***

The first approach is known as the FAST methodology. Its focus is to identify quickly any existing business process constraints. The process begins by finding and enhancing the improvements that would offer the greatest value. This process has been around since the 1980s and resurfaced again in the 1990s re-titled as Express. Today, a number of organizations are using this method as a part of the maintenance life-cycle stage of the process. This is geared towards continued improvement. Operationally, the FAST approach has just two days to meet with managers and decide on what process needs to be changed. Then, an assigned improvement group has approximately 90 days to accomplish the goals set by management (Harrington, Esseling, & van Nimwegen, *Business Process Improvement Workbook: Documentation, Analysis, Design, and Management of Business Process Improvement*, 1997).

This method was considered for this DBIDS project, but due to the scope of the problem and limited resources, it was not practical. Furthermore, the FAST approach typically improves key metrics only 5 to 15% within three months. Our research goals sought a better than 15% improvement to the business process.

**b. *Benchmarking***

The second approach, known as Benchmarking, is at least partly similar to the FAST approach. Similarities end when the method must be compared to similar organizations with better business processes. The goal was to emulate or replicate the good processes of the compared organization and apply them. This approach would look at improving 5 to 20% of the organizations major processes within a six-month period. This would be only a 5% increase from the FAST method (Harrington, Esseling, & van Nimwegen, *Business Process Improvement Workbook: Documentation, Analysis, Design, and Management of Business Process Improvement*, 1997).

**c. *Business Process Reengineering***

Business Process Reengineering (BPR) is considered the most radical and disruptive of the methods being reviewed. “Rather, it is a strategic tool that can be useful, when the organization wishes to achieve dramatic improvement in its performance level (Andersen, 2007).”

The reengineering project is usually initiated by top management who assigns a Process Improvement Team (PIT). These teams are organized to develop a new business process or improve existing processes to be as close to perfect as possible in key performance metrics. However, when it was first introduced, this methodology had negative connotations because organizations adopting this approach were considered it to be “faddish.”

A study undertaken by COBRA (Consultants and Opportunities in Business Restructuring-Analysis) into 100 European companies confirmed that reengineering has become associated with negative aspects of downsizing and cost reduction. (Cook, 1996)

This approach is considered to be costly and time consuming, with only moderate chance of success. However, when applied correctly, this approach has been shown to reduce cost and cycle time by 60–90% and error rates between 40-70% (Harrington, Esseling, & van Nimwegen, Business Process Improvement Workbook: Documentation, Analysis, Design, and Management of Business Process Improvement, 1997).

Again, this approach was not chosen for the DBIDS improvement project due to its emphasis on obliterating the existing system and starting from a blank sheet. The concept of starting from a blank sheet meant that there was a blank check ready to fund a new system. Additionally, DMDC’s current policy would not allow for reconfiguring of their network.

**d. *Process Redesign***

Process Redesign is a subset of the reengineering approach. This method focuses on a single business process. Both processes share similar

steps to attain their objective. Process Redesign is focused on improving the immediate administrative business process performance flow. Typically, the redesign approach is triggered by middle management to address existing IT system performance constraints. When process redesign is fully implemented, it has the potential of obtaining between 30 to 60% improvement. Additionally, it allocates between eighty to a hundred days depending on the complexity of the improvement project.

Ultimately, employing Process Redesign was the most qualified approach to address the research goals motivating this thesis. Figure 3 illustrates the series of steps that were taken to achieve the redesign objective. The steps are to establish a baseline model, analyze it, recommend a surrogate process, and finally implement the redesign. These steps were facilitated by a Business Process Management (BPM) software tool developed by Savvion. The Savvion process modeler software allowed the project team to quickly develop functioning process models, simulate different scenarios, and analyze the processes. Regular analysis of the business process without the BPM software would have been laborious and impractical.

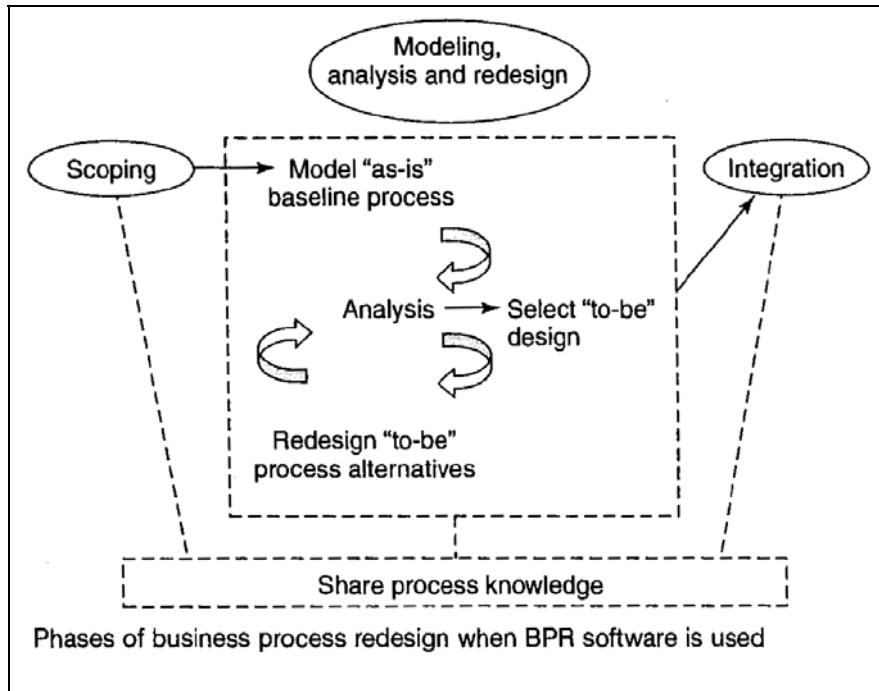


Figure 3. Redesign Process (El Sway, 2001)

## B. MODELING AS-IS BASELINE PROCESS

In Chapter II, an analysis of the As-Is process was performed. However, more emphasis was placed on the external or output-based measurements than internal measurements. In order to evaluate the process flow through Savvion, a review of the business process was conducted. This section focuses on establishing a baseline As-Is model with input from the interviews and questionnaires conducted previously (Table 2).

Measurement parameters in the As-Is process (Appendix E—Process Simulation) were established as a baseline to compare different model designs being evaluated for the purpose of redesigning the process. Measurements were taken after each evaluation of the process.

Measurement	Savvion Definition
Process Time	Time: from arrival of the request to the completion of the request
Performers Queue length utilized	Percentage of utilization
Bottleneck	Name of the processes
How much wait time percentage of total time	Percentage
Cost	Cost in resources

Table 2. Measurement Parameters

### 1. As Is Process Description

The process began with the Manager requesting a document from the Operations PM. The Operations PM would create a report and then send it back to the Manager. If the report documents were inaccurate, the Operations PM would request rework from the Regional PMs. The Regional PMs would then update the spreadsheet and resubmit. The Operations PM would then have to consolidate all of the reworked data and then submit it to the Manager. The Manager also performed this rework loop in the business process by validating the report's accuracy. Once the report had been analyzed, the process ended.

#### a. *Initial Configuration of Savvion*

Utilizing the Savvion software to accurately replicate the As-Is process can be challenging in regards to configuring the parameters. Close attention was placed on insuring that the right individual activity work times were represented. The project team had assumed that initial responses from PMs about the average work time for an activity were correct. However, most parameters were relatively adequate to accurately simulate the As-Is process.

In the initial set up of the Savvion model (Appendix E—Process Simulation), each stakeholder was separated by their responsible area to define boundaries and where handoff occurred. These were key areas for measurement opportunities. Key configuration parameters, assigned to stakeholders, are listed below.

Parameters for the As-Is Model:

- Manager (1 @ \$50/hour)
  - Request Reports (work time 2 min)
  - Email Request (work time 10 min)
  - Review Report (work time 5 min)
- Any Operations Program Manager (2 @ \$40/hour)
  - Find Data for Report (work time 1 hr with deviation of 20 min)
  - Review Report (work time 5 min)
  - Create Report (work time 10 min)
  - Email Report (work time 10 min)
  - Email Request (work time 10 min)
- Any Member Regional Program Manager (5 @ \$40/hour)
  - Update Report (work time 2 hr with deviation of 20 min)
  - Email Updated Report (work time 10 min)

## 2. As-Is Process Improvement Options

Given the measurements from the simulation (Appendix E—Process Simulation) Savvion presented a clearer picture of where issues existed within the As-Is system. Although some issues were evident prior to this redesign analysis, it did not provide the numerical rigor we sought. Incorporating numerical values to represent the delta within the system strengthened the argument for a recommended solution.

The object of this section was recommending a possible candidate redesign (To-Be) for future development. At that point, To-Be measures were compared to the previous As-Is process measurements. The first discrepancies that were apparent in both the external and internal analysis were bottlenecks that existed within the Operations PM's area of responsibility. The Operations

PMs activities were backlogged. Additionally, Operations PMs utilized 93% of the time. This was the culmination of both Operations PMs reworking a spreadsheet system plagued with errors. Maintaining that percentage of utilization was not a practical option. The project team had to design a To-Be system that would not just mitigate the above discrepancies but those that were listed in the As-Is simulation (Appendix E—Process Simulation

The project team agreed that reducing the amount of rework conducted by both the Operations PMs and Manager, would significantly reduce the bottleneck. However, the only option that was available was to replace the As-Is system with a system that would mitigate the constraint issues.

### **3. To-Be Process Description**

The following section discusses the project team's redesign of the system with the intent of replacing the As-Is process. The To-Be model begins with the same request for a document report initiated by a Manager. A Manager uses a DBMS to obtain information. The DBMS allows a Manager to view reports quickly. A Manager has limited access to change any reports. A Manager has to request updates in the same manner as the As-Is system. The Operations PM updates the report and sends a return Email message stating the update is complete. Lastly, in the rare case Regional managers have to update the DBMS, they could quickly update their particular information and respond via Email stating that the update was complete. The To-Be decision process was configured for a higher accuracy than the As-Is decision process, thereby eliminating the necessity of some rework.

#### **a. To-Be Initial Configuring of Savvion**

In this section, the following parameters were configured in the Savvion software.

- In Parameters for the To-Be Model:
  - Manager (1 @ \$50/hour)
  - Request Report (work time 2 min)
  - Access the DBMS (work time 2 min)
  - Email to Update (work time 10 min)
- Any Operations Program Manager (2 @ \$40/hour)
  - Update Data (work time 30 hr with deviation of 5 min)
  - Email Project Managers (work time 10 min)
  - Email Manager Update Complete (work time 10 min)
  - Email Request (work time 10 min)
- Any Regional Program Manager (5 Regional manager @ \$40/hour)
  - Update Site Information (work time 5 min)
  - Email Updated Complete (work time 10 min)
- DBMS
  - DBMS (work time 2 min)
  - DMBS2 (work time 2 min)
  - DBMS3 (work time 2 min)

#### **4. Comparison of the As-Is and To-Be Systems**

This section focuses on the comparison between the As-Is and the To-Be models. In the previous section, the project team could only reconfigure the rework process flow if the As-Is system were replaced. With the help of the Savvion software, the project team was able to quickly construct a To-Be model with some assumptions made concerning the parameters. The results from the To-Be were dramatically improved (Appendix E—Process Simulations). The To-Be process illustrated how the rework was partly eliminated. In addition, the cycle time was reduced from five days to only 16 hours. This was a significant performance improvement.

### C. SUMMARY

In the analysis, the redesign approach was the best approach considering the scope of the Process Redesign objective. The project team followed the Redesign Process (Figure 3). The first step of the process was to design an As-Is model baseline using the Savvion software. The As-Is model internal parameters were configured and simulated by the Savvion software. The project team identified what measurement parameters were the most important to observe. Once the As-Is model was analyzed, the project team then recommended a To-Be model that was placed through the same rigorous analysis as the As-Is model. The To-Be model had drastically improved the business process (Table 3) (Table 4).

As-Is Measurement	Value
<b>Process Time</b>	Time: 5 days 46 minutes
<b>Performers Queue length utilized</b>	93.41% Operations PM 10% Manager
<b>Bottleneck</b>	Find Data for Report (33.51) Create report (18.48)
<b>How much wait time percentage of total time</b>	93%
<b>Cost of Resource</b>	\$17,451.84

Table 3. As-Is Model Measures

To-Be Measurement	Value
<b>Process Time</b>	Time: 16 hours 14 minutes
<b>Performers Queue length utilized</b>	33% DBMS .2% Regional PMs
<b>Bottleneck</b>	DBMS (0.9)
<b>How much wait time percentage of total time</b>	6.2%
<b>Cost of Resource</b>	\$273.33

Table 4. To-Be Model Measures

## **IV. DBIDS DATABASE DESIGN & DEVELOPMENT**

### **A. DBIDS DATABASE DESIGN**

With the modeling of the proposed To-Be process completed, the project team recommended the integration of a DBMS. With this recommendation and its concurrence with the stakeholder's requests, the final phase of integration in the Business Process Redesign began with the designing of the DBMS. To begin this phase of integration, the project team had to begin designing this proposed DBMS.

A key concept in designing an effective DBMS application is to ensure that the development team maintains a data-oriented approach (Ponniah, 2003). From the beginning of work on the project, an effort was made to maintain this focus on the data that was being represented in the As-Is system. With this focus, the project team concentrated on the DBMS and its proper representation of the data, and secondly, on ensuring that the Graphical User Interface (GUI) provided the access to that data in such a way as to replicate the business processes accurately.

### **B. DEVELOPMENT LIFE CYCLE**

The Database Development Life Cycle (DDLC) is an iterative process with many phases that can be performed in parallel. The six phases in this cycle are planning, feasibility, requirements definition, design, development & implementation, and maintenance & growth. This project, however, had the additional requirement of the co-development of a GUI. The life cycle process for a GUI closely follows the same phases with specific differences discussed later in the phases. This chapter will discuss the DDLC process up to the just prior to the implementation of the project.

## **C. PLANNING**

The planning phase is used to identify and address the scope, key issues and assumptions, expectations, data requirements, implementation strategy and creation of a tentative schedule. Many of these items were discussed, and agreed upon, with few adjustments made after the project team began development.

### **1. Scope**

The scope of the project was to create an application that would partially or fully transition the DBIDS data from a spreadsheet to a relational database.

### **2. Key Issues**

Several key issues added hurdles in the project.

- The project would have to be compatible across several different regional offices. With PMs located on the West Coast, East Coast, Europe, Korea and a yet-to-be-established office in the Middle East, this would require the project to adjust to the lowest common denominator when identifying the application and communications tools with which to implement the project.
- The project would require that all of the stakeholders fulfilling a database administrator's role would need an understanding of relational databases.

### **3. Assumptions**

- All of the data contained in the As-Is spreadsheet would be transitioned to the project.
- The DMDC Stakeholders assumed that they would have the project team at hand to address issues and help guide the end users in transitioning to the completed project.

#### **4. Expectations**

The stakeholders expected an application that would:

- Improve accessibility to permit multiple end users in multiple locations to make near-real time updates to the project.
- Improve flexibility in preparing end user reports from these data.
- Improve accountability and traceability for each update to the database.
- Improve accuracy in making data updates.
- Improve scalability of the data management to accommodate more end users and more installations within this system.
- Improve security by limiting an end user's ability to view, add, delete, or edit the data.

#### **5. Data Requirements**

Early in the project, a special data requirement was identified during interviews with DMDC stakeholders. This requirement necessitated that the DMDC Stakeholders be separated into groups to establish data ownership. It was requested that the Regional PMs would only have access to view and update data that was directly associated to that PM's responsibilities. Additionally, Managers would have the ability to view all of the data but have no modification abilities. This business rule was not implementable in the As-Is spreadsheet.

#### **6. Tentative Schedule**

During the initial meetings with NPS Faculty, the project team and the stakeholders, a prototype demonstration was planned for November 2008. The objective of the prototype demonstration was to faithfully model the As-Is spreadsheet in a relational database environment. Feedback from this

demonstration would drive adjustments to the database structure prior to the next milestone. Basic functionality would include create, read, update and delete functions, with rollup reports from the As-Is spreadsheet.

Stakeholders requested that the project team transition from the As-Is system to a live DBMS by January 2009. This live DBMS would have a current snapshot of the data from the As-Is spreadsheet. With this transition the stakeholders would no longer be required to maintain the As-Is spreadsheet.

## **D. FEASIBILITY**

The feasibility phase is a study of three separate areas: technological infrastructure, operational skills, and economics. This study finds the stakeholder's current capabilities in these areas, or the stakeholder's ability to provide additional capabilities in these areas. Sometimes both occur.

### **1. Technological Infrastructure**

The feasibility review of the technological infrastructure focuses on two areas and their adequacy for a proposed DBMS. As mentioned in the planning phase, this was a key issue. Due to the geographically dispersed regional offices, this caused a serious limitation on the options available for the DBMS. The common hardware across the offices was IBM-based personal computers, connected over a Wide Area Network, running Windows XP, and Microsoft Office 2003.

### **2. Operational Skills**

Operator skills were identified as an issue. The project team recognized that for the continued growth or expansion of the project, anyone acting as a Database Administrator would require an understanding of relational databases and the Structured Query Language (SQL). With this knowledge, an administrator could add or edit data fields, forms, or reports, as business rules and processes were added or changed.

### **3. Economics**

For this project, economics acted as a limiting factor. There was no allotment of funding for the development of this project.

With the limiting factors in infrastructure and economics, a selection of Microsoft Access 2003 rather than Microsoft Access 2007 and SharePoint was made. However, this final decision was not made until after the initial prototypes had been demonstrated and development had already begun because of a subsequent discovery. Several of the Regional Offices networking systems and computer services falling under the control of external organizations could not provide the access to the newer software versions.

## **E. REQUIREMENTS DEFINITION**

The requirements definition phase drives the design and implementation of a DBMS. In this phase, a close study is required of the business rules, interactions, procedures, recorded data, and personnel. Additionally, this phase is typically revisited throughout the development and implementation phases to review and adjust defined requirements as needed. The project team conducted this phase using three methods: interviews and surveys of stakeholders, an As-Is process review, and rapid prototyping.

Detailed project requirements were limited in the beginning because the stakeholders did not have a clear vision of what they wanted as a final product. This required the project team to revisit design phases, creating examples of possibilities, modeling those examples and then adjusting the software upon review by the stakeholders.

### **1. Detailed Review of the As-Is Spreadsheet**

A detailed review was completed of the As-Is spreadsheet. To conduct this review, a greater understanding of the data was required; this was accomplished by conducting several interviews with the stakeholders. The project team identified the following about the existing system:

- The As-Is spreadsheet had numerous copies in various states of updating, located on several machines and shared location causing a lot of confusion in understanding which file had the most current and accurate information.
- There were inconsistent data in the same fields across the various sites. Wrong data types in a field, such as “Yes” or “FY09” in a date field, or notes along with a date rather than having the notes consolidated in the notes field.
- DMDC administrators had no ability to accurately identify who or what changes were being made or if any were made at all.
- DMDC administrators had to spend much of their time correcting errors often creating multiple versions of reports.
- The As-Is system made it very difficult see a clear picture about the individual status of site. This difficulty was caused by the over 100 columns of information with possible duplicate entries of sites on over 1000 rows spread across nine different tabs.
- Finally, the As-Is spreadsheet lacked any security, allowing anyone to make changes or remove copies of the files.

## **2. Identified Requirements for the Project**

The decision concerning the particular DBMS application and version was a major issue in the early stages of the project. The original intent was for the project to transition from the As-Is spreadsheet to a Web-based DBMS, using Access 2007 within a SharePoint environment. However, software restrictions within the DMDC network environment mandated rolling back to Access 2003. These restrictions required the Access 2003 database to reside on a drive connection over the WAN. The remaining requirements were for the final product to:

- Contain all of the data in the “current” As-Is spreadsheet.
- Improve accessibility to permit multiple users in multiple locations to make near-real time updates to the database.
- Improve flexibility in preparing user reports from these data.
- Improve accountability and traceability for each update to the database.
- Improve accuracy in making data updates.
- Improve scalability of the data management to accommodate more users and more installations within this system.
- Improve security by limiting who can view or edit the data.
- Contain an initial set of reports, covering the summary worksheets from the As-Is spreadsheet.
- Restructure the As-Is spreadsheet data, cleaned up to between second and third normal form.

## F. DESIGN

The design phase transforms the requirements collected in the previous phase into a plan for the project. There are two parts to this phase the logical design and the physical design. Additionally, in this section, aspects of the GUI design stages will be addressed.

### 1. Logical Design

The logical design is a high-level view of all the data to be contained within the DBMS. In this design phase, the data is divided into separate entities. This phase identifies the relationships between each of these entities within the database (Figure 4). This information was gathered through the review of the As-Is spreadsheet and the interviews with the stakeholders. From the beginning, it was readily identifiable that the main entity to be represented would be the Sites entity. However, after gaining a better understanding of these data, there were additional entities that needed to be introduced in the process of

normalizing the data. As the entities were listed, their relationship with all other entities had to be defined and revisited as more entities and relationships were identified.

During this analysis, the team identified a learning point concerning a full understanding of the data. Early on, an error was made by the project team in assuming that unit commands would hold a significant role in the relationships of the sites, Service Branches, COCOMs, and locations. This assumption led to a difficult update process late in the development phase in order to remove unnecessary complexity.

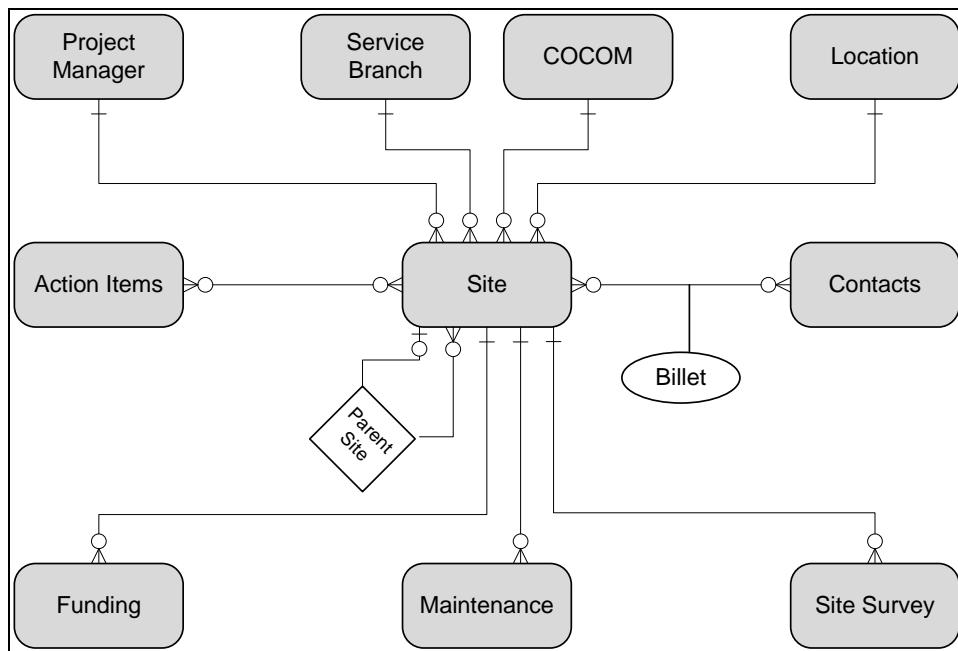


Figure 4. Entity Relationship Model

The primary entity was the individual site (Figure 4). All other additional entities are in some way related to the site entity. Through the reviews mentioned in the requirements definition phase, five relationships were identified for simple entities to aid in sorting and filtering. By placing these data values in separate entities, the project could enforce strict adherence to a specific set of data values and create list controls within the GUI later.

- Project Manager contains a restricted list of all possible PMs, a simple way to show which PM is assigned to a specific site. A single one-to-many relationship was identified with the Sites entity.
- Service Branch contains a list of all the possible service branches that a site can belong to, such as Army, Navy, Marine, Air Force, Joint, and State Department. A single one-to-many relationship was identified with the Sites entity.
- COCOM contains a list of all the possible combatant commands around the world that a site can belong to, such as NORTHCOM, EUCOM, PACOM, CENTCOM, and AFRICOM. A single one-to-many relationship was identified to the Sites entity.
- Location contains a list of all the possible locations around the world where a site can be established. This list contains the location name, the region or state, country and an additional identifier specifying whether a site is within the continental United States or not. A single one-to-many relationship was identified to the Sites entity.
- Parent Site Relationship identifies whether a site falls under the direct supervision of another site. A single circular one-to-many relationship was identified with the Sites entity.

There were three history type entities identified for an individual site. Each of these entities could have zero, one, or more entries per site, allowing a viewing of the history of those areas.

- The Funding entity contains a history of funding information for a particular site and can be expected to show annual funding data throughout the life of a site. A one-to-many relationship was identified from the Sites entity and this entity.
- The Maintenance entity contains a history of maintenance issues associated to a site. A one-to-many relationship was identified from the Sites entity and this entity.

- The Site Survey entity contains a history of the surveys conducted for a site, usually completed prior to and during the establishment of a new site allowing the DMDC Stakeholders to track the timeliness of steps taken during this phase. A one-to-many relationship was identified from the Sites entity and this entity.

The last two entities were identified during the implementation phase, and subsequently added to incorporate the data contained in two additional spreadsheet files containing that data. These two entities can be associated with one or more sites, and a site can be associated to one or more of each of the entities.

- Action Items contain DMDC Stakeholders' office action items and their descriptions. A many-to-many relationship was identified between this entity and the Sites entity.
- Contacts contains all of the contact information, such as the name of the contact, physical address, shipping address, phone numbers and the billet the contact has associated to an individual site. A many-to-many relationship was identified between this entity and the Sites entity.

Once each entity was identified, the individual attributes that each entity would address had to be identified and defined. Here again the review of the As-Is spreadsheet and the interviews completed during the requirements phase provided the project team with sufficient understanding of the data to move forward.

## **2. Physical Design**

The physical design takes the logical design and moves the abstract ideas to an actual representation of the data in a specific hardware or software. From early in the project, it was understood that the project would need to be

developed using either Access 2003 or Access 2007. With this constraint identified, the project team created a representation of the Logical Design within Access 2007 (Figure 5).

As the entities were created as tables, the individual entity attributes were created as fields within each table. This creation was accomplished quickly with a fully developed logical design. Some additional tables and fields had to be created to assist in the representation of the logical design. Primary keys and foreign keys had to be added to relate tables to one another. To represent a many-to-many relationship properly, a link table had to be established between the two entities. These link tables contained the key from each of the two entities and any attributes associated to the relationship. An example of this can be seen with the link between a contact and a site. An additional field of billet is located in the SitesToContactLink table in addition to the primary keys of a contact and a site.

Additionally, there were six tables created along with their relationships to allow for the backing up of a deleted site and its linked data. This functionality was requested after the project was implemented when a change was made in the deletion capabilities of a Regional PM. In addition to this functionality, the project team added the ability for an Operations PM to review or recover any of the deleted site information. The ability to maintain the oversight and accountability of a deleted site necessitated the addition of the backup tables.

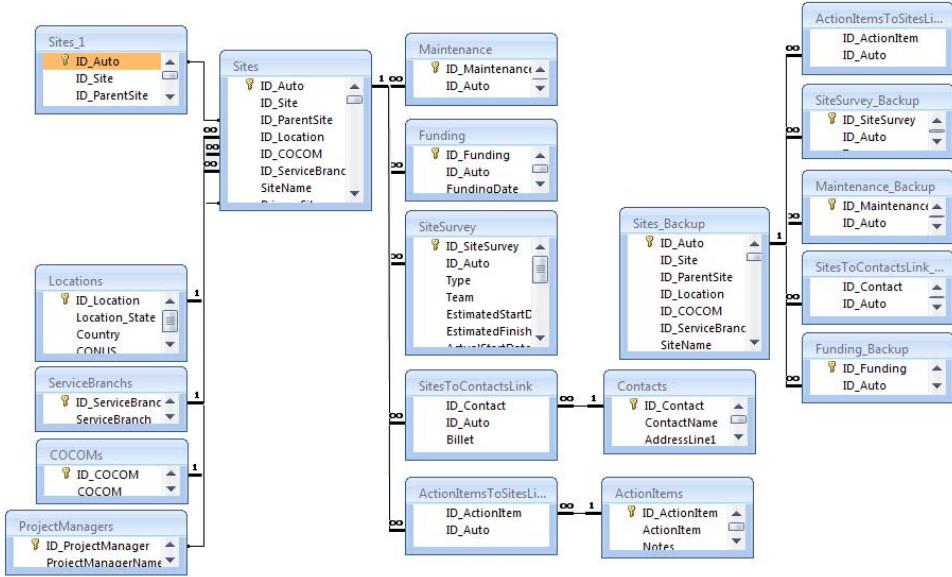


Figure 5. Access Key-based Relational Model

### 3. Graphical User Interface (GUI) Design

In this phase of the design, the GUI portion associated to viewing, inputting, and retrieving output from the database had to begin being flushed out. Here again the driving force and primary source of direction was obtained in the requirements definition phase with the review of the As-Is system and the interviews with the stakeholders. Having the ability to view existing examples of hand-created reports greatly aided the project team in designing the GUI. The interviews led to a greater understanding of the data allowing for a more logical grouping of information for review and entry forms. With this knowledge and the listing of entities and their relations, forms and reports were planned to show these aspects of the data.

An additional requirement became evident early in the discussion of designing the GUI. This new requirement was necessitated by a requirement to have at least two separate viewable GUIs; one for the regular end users, i.e. Managers and Regional PMs; and an added view of the administrator type views and functions for the Operations PMs.

After the initial design of the GUI, only one additional input view was requested by the Regional PMs. The view requested was a datasheet view that more closely resembled the spreadsheet type interface, allowing for rapid updating of multiple sites by a user. The requested view was to allow the Regional PMs to more easily transition from the spreadsheet type interface of the As-Is system to the form type views within the Access environment. The items that received the most attention after the initial design were search and filtering capabilities on forms, audit tracking, and the formatting/creation of reports.

#### **4. Prototypes**

Several times during the database and GUI design, a simple mock up of a form, report, or menu was used to convey ideas between the project team and the stakeholders. The use of the prototypes greatly increased the understanding between the personnel on both ends of the design. Additionally, partially working forms or versions of the project could be delivered and tested by actual users. This allowed for key feedback throughout the design and development phases. These feedbacks led to a better understanding of the GUI requirements and identified errors made in the GUI or representations of procedures. The following are a few of the major prototype deliveries:

##### **a. *Version 0. –16 October 2008***

The requested initial prototype, mentioned in the tentative schedule, was delivered, and briefed, on 16 October 2008. With this demonstration, the basic layout of the GUI was modeled. Additionally, the requested Quick Stats Report was modeled, and the stakeholders were greatly impressed.

##### **b. *Version 1.0–4 December 2008***

A functional database was loaded with testing data from an outdated version of the As-Is spreadsheet. All of the procedures used to import

this testing data were saved in preparation for going live and importing the most current data from the As-Is spreadsheet.

**c. *Version 1.11–9 January 2009***

Due to differences in regional software installations discussed in the planning phase, the final decision on the software version was identified and the database was rolled back from Access 2007 to Access 2003. Introduction of the Action Items entity and a “Go-to drop-down” control was added to each form.

This was to be the initial delivery identified in the tentative schedule. However, due to the major changes caused by the roll back and scheduling conflicts within DMDC, the personnel could not be gathered for the initial introduction of the project.

**d. *Version 1.20–1 March 2009***

Capability to filter reports introduced.

**e. *Version 1.21–13 March 2009***

A Functional version that was modeled to all stakeholders including the Regional PMs. This version was opened to all of the stakeholders to view and test the GUI.

## **5. Design Phase Results**

Several documents were created from the initial design phase results and then maintained as changes during the development and maintenance arose.

**a. Key Based Data Model**

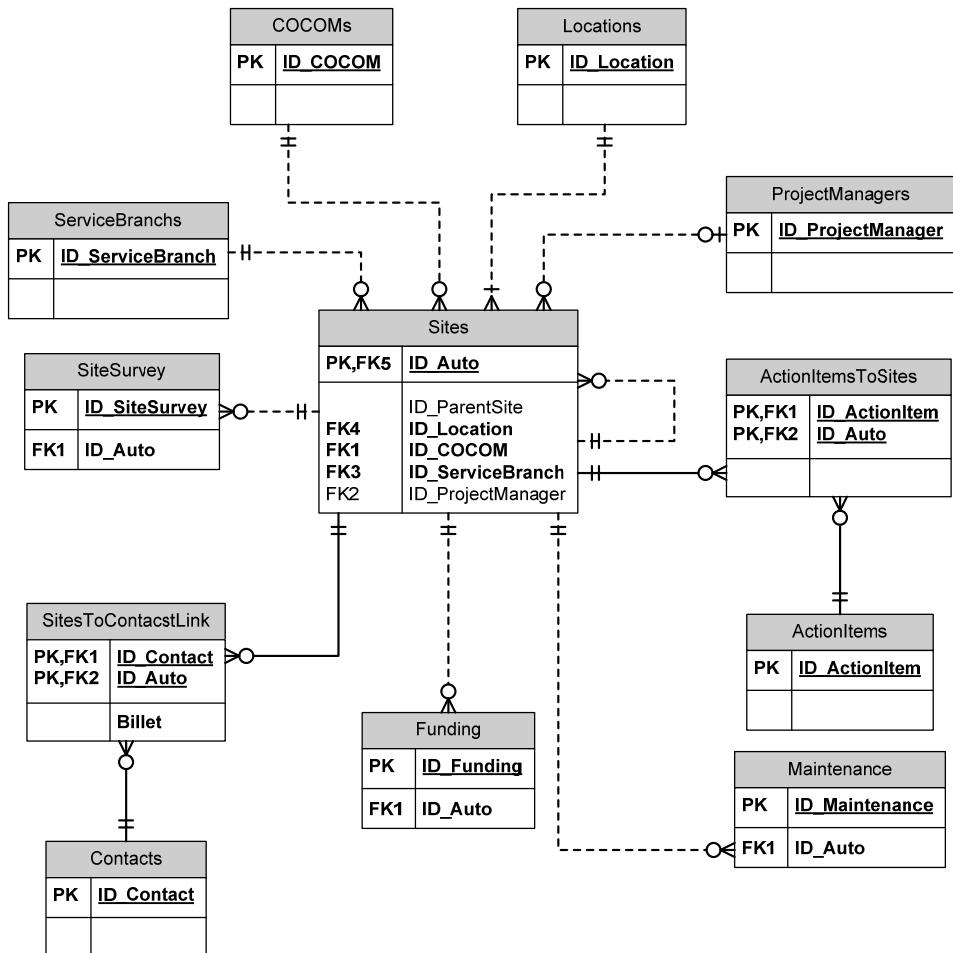


Figure 6. Key Based Data Model

**b. Entity Definition Matrix**

Entity	Business Definition
<b>Sites</b>	Primary entity containing all site specific information.
<b>COCOMs</b>	Combatant Command used to filter and sort sites.
<b>Locations</b>	Geographic Location of a site. Used to filter and sort sites by State & Country.
<b>Project Managers</b>	Project Managers Name.
<b>Action Items</b>	Action Items for DBIDS sites, may cover more than one site.
<b>Maintenance</b>	Listing of maintenance conducted for each site.
<b>Funding</b>	Listing of funding or projected funding for each site.
<b>Contact Information</b>	Listing of contact information for individuals acting in specific roles for a site.
<b>Site Surveys</b>	Listing of surveys conducted for each site.
<b>Service Branches</b>	Service Branches with command over a site. Used to filter and sort sites.

Table 5. Entity Definition Matrix

**c. Stake Holder Glossary**

(See Appendix F—Stake Holder Glossary)

**d. Use Case Glossary**

(See Appendix G—Use Case Glossary)

**e. Social Network**

(See Appendix H—Social Network)

**f. Switchboard Mapping**

(See Appendix I—DBIDS Master Database Help File / Switchboard Mapping)

## **G. DEVELOPMENT**

### **1. DBMS**

Initially, the database system to be used had been identified as Access 2007. Due to the limitations discussed in the feasibility phase, the database system was changed to Access 2003 in the development phase of the project.

#### ***a. Physical Design Created in Access***

The first step in the process of creating the Physical Design in the Access environment was centered on creating tables and the associated fields with the appropriate data types. Once the tables were created, the relationships needed to be created with the appropriate cardinality. Access 2003 has a built in relationship view that is easy to manipulate. However, an understanding of how to represent a specific cardinality within the Access environment is required.

#### ***b. Data Conversion***

One of the issues identified in the As-Is spreadsheet review was the inconsistent use of data types within a specific column. This caused many issues in trying to force improperly formatted the data into the tables within the database. Examples of this could be found throughout the As-Is spreadsheet. The As-Is spreadsheet would have numerous date formats, from “dd/mm/yy” to “dd-mmm-yyyy”. Additionally, there were many instances of date fields containing an ‘X’. This marking was being used to identify “YES, an item had been completed”, without actually entering any date values.

An additional problem was the normalization of the data. This required several columns of information to be read into the database to populate separate tables. Then upon creation of these new tables, the individual primary keys associated to each unique record had to replace the previous text entries creating foreign key entries. An example of this was seen with the creation of the COCOMs table. The possible unique values had to be read from the As-Is spreadsheet; then those values had to populate the COCOMs table, creating a

primary key value for each. Finally, the column associated to the COCOM field in each row of information from the As-Is spreadsheet had to be replaced with this foreign key entry.

Early on in the development, there was the need to have a large test data set within the various versions of the database. This need led the project team to developing stored procedures within Access 2003 to make these conversions. These stored procedures greatly increased the speed in which the spreadsheet data could be fully implemented into the database.

## **2. GUI Development**

The intention of a GUI is to facilitate the visual interaction between the end user and an application, without having an intimate knowledge of the programming languages and connections required to make the various tasks possible.

Since the majority of the interactions with the project would be done by the Operations PMs, the project team guided the GUI design predominantly around these interactions. Regularly meeting with the Operations PMs, allowed the project team to capture processes and the relations of those processes to the actual data being stored. By relating these processes and data, the project team could then ensure like data and tasks were properly grouped and displayed by the GUI to allow for greater usability. Several tools were used within the project to enhance usability and facilitate this interaction.

### **a. *Structured Query Language (SQL)***

To be able to retrieve data from the tables within the database, an understanding of SQL is required. However, the goal of the GUI is to hide this requirement from the end users, this is accomplished through the use of stored SQL procedures. A basic SQL command is used to specify what data is needed, how to create it, and from where to retrieve the data. Within Access 2003, there is the ability to execute SQL commands using a built in visual tool. This tool is

very useful in the execution of simple SQL commands. However, it is not able to model all of the possible SQL commands. An example of this limitation would be the UNION command.

***b. Visual Basic (VB) Coding***

Visual Basic is an event-driven programming language from Microsoft and is the heart of GUI programming within Access 2003. Though many simple tasks can be created within Access 2003 using macros, the macros are not robust enough to accomplish every task required for this project. A majority of the VB coding for this project was centered on the forms within the database and events associated with the user's interactions with those forms. A simple example of this is the clicking of a button. By default, when adding a button control to a form within Access 2003, a function was created in VB, and an entry was created linking the event of a mouse click to this segment of code. From that time on, when the form was opened for use, a mouse click on the button would execute the segment of code associated with it.

***c. Queries***

A Query is a question, written in SQL, from the user to the database. There were almost fifty separate queries created in the project, not including SQL written directly into the VB code dynamically creating queries, nor does it include the numerous queries used to import the raw data from the spreadsheet to populate the database initially. An effort was made to reduce complexity and remove duplicate effort among separate queries. To view the specific information on each query within the project, refer to the Appendix I—DBIDS Master Database Help File. The four basic types of queries used in the project are:

(1) A SELECT query is asking to look at some information within the database. These queries allow the dynamic grouping, sorting, formatting, and or creation of specific data. A majority of the queries written for the project are of this type.

(2) An ADD query allows records to be appended to a table within the database.

(3) An UPDATE query allows for the editing and modification of records within the database.

(4) A DELETE query allows for the removal of records from the database.

***d. Forms***

The forms within Access 2003 are the predominant interaction that an end user will have with the database. A form opens a window within Access 2003 displaying the items associated to it. To view the specific information on each form within the project, refer to the Appendix I—DBIDS Master Database Help File. The following are the most important properties of a form.

(1) The record source points to the query or table from which the form is pulling its information.

(2) The controls of a form are each item contained within the form. The two most used types of controls are the text box and the label. A text box allows for the display of data and or the entry of values. Whereas, a label is just written text usually used to identify other controls.

(3) An event is an interaction between a user and the GUI, such as mouse clicks on controls, changes of data, opening or closing a form. A majority of the time spent working in the project was centered on the coding and handling of specific events.

***e. Reports***

A report can be created and stored, within Access 2003, to output a set format of the information within the database. To view the specific

information on each report within the project, refer to the Appendix I—DBIDS Master Database Help File. The following are the more important properties or capabilities of a report.

(1) The filtering capability of the data for a report can limit the information displayed on the report. Most of the reports are opened using the tblOpenReports data and the frmOpenReport. The use of these two items allowed for the dynamic filtering of sites to shown in a report.

(2) The grouping capability can gather data that is related by a value in a certain field. An example of this would be all of the contacts for a specific site. The grouping on the report can group each contact having the same Auto\_ID for a site. This would allow the report to display the site information one time and then a listing of all the contact information for that site.

(3) The sorting capability is a simple attribute allowing the report to setup the order in which records are displayed. Sorting can be done in conjunction with grouping. Taking the site and contact grouping from above, the report can, after grouping each site's contacts together, sort the sites by location, site name; and then by the contact's name. The only limitation is that the total number of sorts and groupings cannot exceed ten.

#### ***f. Security***

There are four levels to the security used in securing access to the database. The first two are part of the DMDC networking environment. DMDC Stakeholders are required to login to a computer on site with their personnel login information that authenticates their identity to the system. Once this is accomplished, a user would have to be given permission, by network administrators, to access the file location where the database is located on the WAN. The next step in the security deals with the use of a key, to access the encrypted database. A short-cut was created pointing to the location of the Access 2003 application, location of the database and the key used to access the database. Once Access 2003 is running, the user would have to login with

another id and password, unique to the database. Based on the credentials, the database then filters and shows the appropriate information for that user's level of access.

Upon opening the database, there are a specific set of tasks carried out by default. However, there is the capability to do what is called a Shift-By-Pass to skip these steps. Part of the security was the ability of an administrator to toggle this capability. By keeping this bypass capability turned off, the administrators can keep other users from viewing or accessing the inner working of the database. An Administrator can then allow this capability to grant themselves access to the development views of the database.

One limitation to performing updates to forms, reports, tables and queries, is that an administrator needs to have exclusive access to the database to perform these changes. If an administrator has the need, the capability to view what users are currently accessing the database and either warn the users through the database and or immediately disconnect users from the database, allowing an Administrator to then reopen the database in an exclusive state. This technique ensures data integrity and prevents corruption.

#### ***g. Auditing***

One of the original capabilities requested was the ability to perform an auditing of the changes to the site data. This auditing logs the field being changed, the original value, the new value, the user making the change, and when the change was made. This allowed administrators to review all the changes to the sites information. The capability was also added to backup internally any site deletions, allowing an administrator to recover or permanently delete a site. Later, DMDC Managers put this capability to the additional use of viewing the working habits of the PMs.

#### ***h. Menus & Switchboard***

Part of the databases' default processes upon opening is to limit the menu capabilities available to all users and to open the internal switchboard. Part of the opening events for the switchboard form is to identify the access level of the current user and limit the functions available to him on the switchboard. This allows the switchboard to identify whether the current user logged in is an administrator and to display the additional administrator functions if appropriate.

### **3. Testing**

A major portion of the testing was conducted through the prototypes passed to the stakeholders. This allowed a large amount of user feedback not only on errors but on the functionality of the latest implemented portions of the GUI.

### **H. SUMMARY**

Throughout this design and development process, the project team continuously met with stakeholders. Both the stakeholders and the project team made themselves available to field each other's questions, whether to meet for a clarification of data, explanation of business rules, clarification of requirements, or modeling of new capabilities. This interaction greatly sped this design and development process, allowing the project team to reach a point at which the implementation phase of the DDLC could begin having the project go "live" at the DMDC offices.

THIS PAGE INTENTIONALLY LEFT BLANK

## **V. DBIDS DATABASE IMPLEMENTATION**

After the initial study and planning for the DBMS, the project team was able to begin the final phases of the DDLC by completing the last portions of the development, moving on to the actual implementation, and finally to the maintenance & growth. With the project's completion, the project team had to review their success or failures to meet the project's requirements and expectations.

### **A. IMPLEMENTATION**

The actual installation was conducted over a three-day period 3–5 April, 2009. Due to the continuous prototyping throughout the development phase, the process of conducting the installation seemed to be a phase-in installation to the stakeholders. However, the actual installation was a direct cutover.

#### **1. Version 1.22–3 April 2009**

This version made use of the stored procedures from version 1.0 to bring the most recent version of the As-Is spreadsheet data online in the database and marked the last uses of the legacy Excel system. Additionally, this version introduced the user login functionality and dynamic filtering of the data shown depending upon the user logged-in and the filters in the PMFilter table. Three additional reports were created and implemented into the report filtering functions. Lastly, the Contact Information entity was created, and the initial data was imported from those Excel spreadsheets.

### **B. MAINTENANCE & GROWTH**

#### **1. Initial Maintenance and Growth**

As discussed in the expectations, the stakeholders counted on the project team to be available for a period of time to allow for the full integration and refinement of the project.

**a. *Versions 1.2–1.32–5 April–3 August 2009***

Throughout this period, the database was live and PMs were working on the data on a daily basis. The project team began conducting a one–two week update and maintenance iterations. During these iterations, any issues, identified by either the project team or stakeholders, were addressed and corrected. Throughout this time, as new requests asked for additions or modifications to the GUI, the ideas were introduced in small workable pieces to the live version of the project. Lastly, as the project team completed these iterations, Operations PMs were being introduced to the inner workings of the project and shown capabilities afforded to the Operations PMs as pseudo Database Administrators.

**b. *Version 1.33–7 August 2009***

This was the final version delivery conducted along with a question and answer out-briefing between the project team, NPS Faculty, and stakeholders.

**2. Regular Maintenance Requirements**

It was recommended to the stakeholders that Bi-weekly, the following maintenance steps be performed. For detailed description on the method of performing each step, refer to Appendix I—DBIDS Master Database Help File.

- Purging of audit trail entries.
- Compacting & repairing the database.

**3. Database Management Processes**

Some of the periodic tasks of the Operations PMs required specific steps to accomplish. Here again, these steps were explained in detail. For a description on the method of performing each step, refer to Appendix I—DBIDS Master Database Help File.

- Adding new users.
- Changing filters for a user.

- Gaining exclusive access to the database.
- Adding a new option to the database's switchboard.
- Setting up access to the database.

#### **4. Growth**

Throughout the scope of this project, the project team had to closely manage requirements changes and scope creep. These originated from the stakeholders not fully knowing the capabilities possible with a DBMS in the beginning of the planning phase. By not knowing or understanding these capabilities, stakeholders could not know what they wanted until they saw it. After the project had gone live, the stakeholders could then see these capabilities in action, this triggered requests for additions that had not been in the original requirements. However, many of these additional requirements centered on easily created forms or reports and did not require significant redesign of the project. During these additions, it was apparent that to the project team that for the continued growth of this database application, one or more personnel would be required with the following skills.

- An understanding of relational databases.
- An understanding of the Access 2003 GUI associated to creating queries, or an understanding of SQL.
- An understanding of the functionality of the Access 2003 Design environment.
- For major expansions to the system, an understanding of the Visual Basic language and its interaction within the Access 2003 environment.
- Lastly, the information covered in detail within the DBIDS Help File (Appendix I—DBIDS Master Database Help File) would provide a greater understanding of the internal workings of this database and GUI.

Without this individual to aid in adjusting or adding features as business rules changed or were created, this project could quickly become outdated and useless.

### **C. DOCUMENTATION**

In the later stages of the DDLC process, it became evident to the project team and Operations PMs that the creation of a help file would be mandatory for the continued maintenance and growth throughout the project's life cycle. Without this document to act as a reference source for the elements within the project, future end users or developers would have no understanding of the inner relations of these elements.

With this document, future administrators and or developers could readily view and decipher the data structures and processes within the project. These step-by-step directions would cover specific functions, explanations of specific relations or processes, and detailed listings of the data, controls, and events that are particular to this project.

### **D. RESULTS**

After the project implementation was completed and in parallel with the maintenance and growth phase, the project team began assessing specific measures to identify if the project was successful. Many of these measurements had to be done in comparison with the As-Is system. A final task stated in the scope of the project required the project team to identify whether the objectives were met. Had the expectations and requirements initially set out been accomplished? Some expectations and requirements could be answered with a straightforward "yes" or "no;" however, some were centered on end user satisfaction and perceptions of the final product.

The measurements of success of an IS project are a difficult topic of research. While numerous studies have been undertaken both subjective and objective the most widely used single measure of IS success has been user satisfaction. To what extent are you satisfied with the end result? (Grover, Fiedler, & Teng, 1994)

To begin to evaluate whether the product had met these expectations or requirements, the project team polled the end users through questionnaires and direct interviews. Each requirement and evidence showing completion is listed here:

## 1. Requirements

The stakeholders had these requirements for the final product:

- Contain all of the data in the “current” As-Is spreadsheet. The project team was able to show the capability to import the As-Is data in Version 1.0 delivered in December 2008, and again during the actual implementation on 3 April 2009. However, there were pieces of information that did not meet the new data type requirements. A listing of invalid data was provided to the Operations PMs, and with the aid of the project team, each item was identified and conversions made to properly represent the data.
- Improve accessibility to permit multiple end users in multiple locations to make near-real time updates to the project. The project team was able to model the capability with the first prototype in October 2008, and again on 3 April 2009, when the project was fully implemented.
- Improve flexibility in preparing end user reports from these data. The project team was able to show Operational PMs the various capabilities of reports and their finished products again with the first prototype. Additionally, the project team created new reports, modeling the data within the finished product, numerous times throughout the maintenance and growth phase.
- Improve accountability and traceability for each update to the database. The finished product required a user to log into the system and had the capability to track all changes made to a site’s

data, recording not only the old and previous values but also the name of the user making the change, and the time of that change. After implementation, the project team adjusted this to allow for a change in business rules. This rule change allowed a Regional PM to delete a site and its associated data. With this change, the project team had to implement the ability for the Operations PMs, acting as Database Administrators, to recover deleted sites and their associated data.

- Improve accuracy in making data updates. This was identified in three ways, one as a direct result of applying data type restrictions on a data field. In the As-Is system, there was not a way to insure the correct data type was entered. In an indirect way, the implementation of the audit tracking capability gave Managers a way to observe the entry habits and frequency of PMs usage, thereby, providing an incentive to PMs to update information correctly and in a timely manner. Additionally, the GUI provided the PMs a much easier interaction with the data. These last two were brought to the project teams attention in both interview conversations and questionnaire write-in responses.

The system now is more accurate and PMs are more cognizant about keeping updates current. (Stakeholder)

It has made it easier for the Project Managers to update info, making it more likely that it will actually get updated!!! Contributing, of course, to a better database. (Stakeholder).

- Improve scalability of the data management to accommodate more end users and more installations within this system. With the As-Is system, the end users could not access the data simultaneously and this caused there to be many versions of the file being passed back and forth. By developing the project in Access 2003, this

capability was greatly increased. However, there are some limitations of the end product, but those limitations greatly exceed the current status of the data.

- The maximum allowable file size of an Access 2003 database (.mdb) is 2 gigabytes. During development and throughout the maintenance and growth phases the largest recorded file size was < 25 MB, and by utilizing a built in capability of Access 2003 the file size is compacted to <5 MB once every two weeks. The project team addressed this periodic routine in Appendix I—DBIDS Master Database Help File.
- The maximum allowable number of field per table is 255. After the normalizing the data, the entity with the most attributes was the Site entity. After development, the table containing this entities attributes contained only 72 fields of data.
- The maximum number of objects in a database is 32,768. Currently, there are 22 tables, 44 queries, 34 forms, 14 reports, and 1 VB module.
- The maximum allowable number of concurrent users for an Access 2003 database is 255. Currently, there are approximately 25 different users accessing the end product.
- Improve security by limiting an end user's ability to view, add, delete, or edit the data. The project required each user to login to the system, with that login the system would then look at the privileges for that individual user login. In addition to these privileges, Database Administrators had the capability to limit a user's access further with filters. With the filters, the GUI could

dynamically create Record Source entries for the individual reports and forms that a user was trying to access.

- Contain an initial set of reports, covering the summary worksheets from the As-Is spreadsheet. The report provided in the As-Is system, was a “Quick Stats Report”. This capability was modeled in the prototype demonstrated in October 2008; however, throughout the development and maintenance growth phases additional reports were requested. The project team developed eight additional reports, and further developed these reports for each to have the capability to dynamically filter depending on the user’s filters discussed above and permitting users to further filter the reports within their responsibility.
- Restructure the As-Is spreadsheet data, cleaned up to between second and third normal form. This was accomplished during the design phase prior to the database being developed. By applying, the processes of normalizing data that were taught throughout NPS’ ITM curriculum and further discussed in detail in the reference (Ponniah, 2003).

## **2. Customer Satisfaction**

Through questionnaires and interviews, The project team was able to measure customer overall satisfaction in several specific areas. Of the available answers, Definitely, Probably, Not Sure, Probably Not and, Definitely Not, the results were as follows:

- Did the product help end users manage their job?
  - 57% Definitely
  - 43% Probably
- Can the end users retrieve relevant data from the product more easily than with the As-Is system?

- 86% Definitely
  - 14% Probably
- The end users rated the product as 100% satisfied or very satisfied with functionality, accessibility, security, and accuracy.
- When given an opportunity to identify anything that caused them frustration with the product, the end users had no response.
- Would the end users recommend the product?
  - 43% Definitely
  - 57% Probably
- Lastly, when separate stakeholders were asked if the product provided more security and a separate question asking if it provided more efficiency and effectiveness, the response was “Absolutely!”

## **E. SUMMARY**

This Chapter has covered the implementation phase of an Access DBMS. Arguably, this implementation phase can be difficult given that scope creep often arises in projects of this nature. Realistically, IT projects such as this are not implemented without some form of scope creep present. The objective is to manage the scope creep. project teams were very cognizant about making every effort to quickly refine minor changes to requirements. These requirements were then documented for each new iteration completed. Once requirements were satisfied, the system was authorized to go live. In addition to implementation phase, a maintenance life cycle was considered important due to the dynamic nature of the DBMS. Finally, measures were taken from the second questionnaire, interviews, and compared against the As-Is questionnaire results. To-Be system indicated improvements had clearly made the system more efficient and effective.

In Chapter VI, we will attempt to discuss the recommendation for future work for this study.

THIS PAGE INTENTIONALLY LEFT BLANK

## **VI. SUMMARY, CONCLUSION AND FUTURE RESEARCH**

### **A. SUMMARY**

The Defense Manpower Data Center (DMDC) manages the Defense Biometric Identification System (DBIDS). DBIDS captures personal and biometric information to manage DoD-wide access, control, and personnel accountability. DBIDS equipment is installed globally and managed by a central office on the Monterey Peninsula. Program managers track data about the numbers and type of equipment installed at each site. Program managers were tracking DBIDS data using a single Microsoft Excel workbook comprised of several, interlinking worksheets (DBIDS Master Plan Spreadsheet). Data updates were error-prone and difficult, requiring close coordination to keep the number of “current” versions of the spreadsheet to a minimum. This thesis initially focused on reviewing the business rules and processes surrounding DBIDS document, and then transitioned into designing, developing, and implementing of a relational database solution to improve problem areas identified during the initial review. After implementation of the database, this thesis explored the effects of making such a change within an organization. This was attempted by identifying and measuring changes in performance and accuracy of the system, by measuring pre- and post- user satisfaction through the qualitative methods of questionnaires and interviews, and finally using this analysis to improve the project through maintenance and growth iterations.

### **B. CONCLUSION**

The As-Is system domain was defined as an administrative process for managing DBIDS. The project team conducted interviews and administered questionnaires to establish a baseline for the As-Is system. The results of this review found the As-Is system core processes were neither efficient or effective. The core process of the As-Is system was to record and report the status of DBIDS operations. The project team applied a BPI approach to analyze,

improve, and mitigate the deficiencies with a more streamlined process. Each BPI approach was different considering the purpose and objective of any project.

Of the four leading BPI approaches, FAST, Baseline, Business Process Reengineering, and Business Process Redesign, the project team considered Business Process Redesign to be the best fit. The process focused on improving the performance of the system by 30%.

For the Business Process Redesign approach to be effective, certain steps coupled with the aid of Savvion software were utilized. Using this software, the project team conducted a rigorous analysis of the internal processes. The software enabled the project team to identify performance constraints.

The results of the Savvion analysis indicated that there were constraints within two activities which existed in the Operations PMs activities. These process had considerable rework indicating that time was being wasted to correct errors. The problem of rework was centered on the lack of collaboration capabilities of the As-Is spreadsheet. With this change modeled in a To-Be process, the Savvion software indicated significant improvements.

The project team then shifted its focus toward the replacement of the As-Is spreadsheet with a fully functional DBMS. Utilizing the Database Development Life Cycle process, the project team was able to plan, identify feasibility, define requirements, design, develop and implement, and finally maintain and manage the growth of a DBMS.

Finally, the project team conducted interviews and administered questionnaires to identify and measure the results of applying this change within the process and whether the DBMS had met the requirements. The final results showed a successful change to the business processes, with an application that could meet the current business requirements and had the capability for continued growth. This change allowed the DoD organization to improve an internal process' effectiveness and efficiency at no cost. Additionally, the project team successfully implemented an IT solution in a real world environment.

The project team found issues common to many IT projects. For example, the customer was able to give only vague requirements for the desired end state. However, the project team was able to deal with this through the use of rapid prototyping. In essence, rapid prototyping allowed over thirty iterations to quickly be developed and tested. Successfully tested iterations helped refine the user's requirements.

### **C. FUTURE DEVELOPMENT OR RESEARCH**

While the current implementation met all of the initial requirements identified in the earlier analysis, during the maintenance and growth phase additional requests were made and could not all be accomplished. One of these requests was the ability to attach to a site record any electronic files that may be associated. This capability is not available with Access 2003; however, Access 2007 does have this capability but is limited to an overall 2 GB database file. Upgrading to Access 2007 will not accomplish this request due to this limitation. With the Access 2007 limitation, a site would only allow for an average of 4 MB of attachments. By implementing a more versatile DBMS allowing larger file storage and association, this could aid DMDC in file management concerning site documentation. This type of application could further aid in the reduction of storage space taken up with redundant backup or "current" files spread across their offices throughout the world.

Additionally, to meet the expansion of the DBIDS program within DMDC, future analysis and implementations should focus on a Web-enabled presence or a secure collaboration mechanism beyond the current network shared drive. A Web-enabled database could allow the remote sites to access with a lower amount of bandwidth being utilized. With this type of implementation, the remote sites could be established with a front-end application or use a Web browser to perform the GUI functions locally rather than over the network.

Finally, senior leadership may want to consider integrating this equipment management database with other DBIDS spreadsheets or databases dealing with equipment maintenance, serial numbers, and other day-to-day operations data. This could eliminate much of the hand-computation involved in updating the current DBIDS fields, permitting managers to spot check figures rather than engage in error-prone data entry. Through this integration, DMDC may also be able to implement a robust strategic tool such a Decision Support System. Stakeholders could use this tool to build a consensus to initiate the right course of action. With a framework centered on a Decision Support System coupled with strong business rules, DMDC could more readily forecast potential problem areas and make more proactive decisions vice reactive ones.

## LIST OF REFERENCES

Andersen, B. (2007). *Buisness Process Improvement Toolbox* (2nd Edition ed.). New York, New York: Asq Quality Press.

Bently, L. D., & Whitten, J. L. (2005). *Systems Analysis and Design Methods*. New York: McGraw-Hill/Irwin.

Bhatt, G. D. (2000). Exploring the relationship between information technology, infrastructure and business process re-engineering. *Buisness Process Management Journal* , 6 (2), 139–163.

Cook, S. (1996). *Process Improvement: A Handbook for Managers*. Aldershot Hampshire, England: Gower Publishing Limited.

Cooper, A. (1995). *About Face—The Essentials of User Interface Design*. Foster City, California: IDG Books Worldwide, Inc.

Defense Biological Identification System (DBIDS). (2006, March). *DBIDS Users Conference—2006*. Retrieved September 2009, from <https://www.dmdc.osd.mil/dbids/index.html>

Department of Defense. (n.d.). *Defense Link*. Retrieved September 2009, from <http://www.defenselink.mil/privacy/notices/osd/dmdc01b.shtml>

EI Sway, O. A. (2001). *Redesigning Enterprise Processes for E-Business*. New York: McGraw Hill Higher Education.

Grover, V., Fiedler, K., & Teng, J. (1994). Exploring the success of information technology enabled businessprocess reengineering. *IEEE Transactions on Engineering Management* , 41 (3), 276–284.

Harmon, P. (2003). *Business Process Change: A Manager's Guide to Improving, Redesigning, and Automating Processes* (The Morgan Kaufmann Series in Data Management Systems). San Francisco, California: Morgan Kaufmann Publishers.

Harrington, H. J. (1991). *Business Process Improvement: The Breakthrough Strategy for Total Quality, Productivity, and Competitiveness*. New York: McGraw-Hill.

Harrington, H. J., Esseling, E. K., & van Nimwegen, H. (1997). *Business Process Improvement Workbook: Documentation, Analysis, Design, and Management of Business Process Improvement*. New York, New York: The McGraw-Hills Companies, Inc.

Jackson, W. (2009, March 23). *GAO: Better use of IT could help DHS reduce backlog of FOIA requests*. Retrieved September 2009, from Government Computer News: [http://gcn.com/articles/2009/03/23/dhs-foia-improvement.aspx?sc\\_lang=en](http://gcn.com/articles/2009/03/23/dhs-foia-improvement.aspx?sc_lang=en)

Kock, N. (2007). *Systems Analysis & Design Fundamentals: A Business Process Redesign Approach*. Thousand Oaks, California: Sage Productions, Inc.

McConnell, S. (1993). *Code Complete—A Practical Handbook of Software Construction*. Redmond, Washington: Microsoft Press.

Melan, E. H. (1993). *Process Management: Methods for Improving Products and Service*. New York: McGraw-Hill.

Mertler, C. A. (2009). *Action Research: Teachers as Researchers in the Classroom*. Thousand Oaks, California: Sage Publications, Inc.

Microsoft. (2006, October 11). *Microsoft Help and Support—ACC2000: Sample Functions to Check User and Group Information*. Retrieved September 2009, from <http://support.microsoft.com/?kbid=210331>

Microsoft. (2006, October 11). *Microsoft Help and Support—How to create an audit trail of record changes in a form in Access 2000*. Retrieved September 2009, from <http://support.microsoft.com/default.aspx?scid=kb;en-us;197592>

Microsoft. (2007, July 3). Microsoft Help and Support—How to determine who is logged on to a database by using Microsoft Jet UserRoster in Access. Retrieved 2009, from <http://support.microsoft.com/?id=285822>

Moore, J. (2004, February 22). *Process re-engineering back again*. Retrieved September 2009, from [http://fcw.com/articles/2004/02/22/process-reengineering-back-again.aspx?sc\\_lang=en](http://fcw.com/articles/2004/02/22/process-reengineering-back-again.aspx?sc_lang=en)

Ponniah, P. (2003). *Database Design And Development—An Essential Guide for IT Professionals*. Hoboken, New Jersey: John Wiley & Sons, Inc.

Weerakkody, V., & Currie, W. (2003). *Integrating Business Process Reengineering with Information Systems Development: Issues & Implications. Lecture Notes in Computer Science*, 1019.

Wetherbe, J. (1988). *Systems analysis and design: Traditional, structured, and advanced concepts and techniques*. (3rd Edition ed.). Saint Paul, Minnesota: West Publishing.

Wysocki, R. K., & McGary, R. (2003). *Effective Project Management: Traditional, Adaptive, Extreme* (3rd Edition ed.). Indianapolis, Indiana: Wiley Publishing, Inc.

THIS PAGE INTENTIONALLY LEFT BLANK

## APPENDIX A—QUESTIONNAIRE RESULTS

DBIDS Master Plan Spreadsheet (As-Is) Questionnaire						
Q1. Do you use the DBIDS Master Plan Spreadsheet?						
Answer Options						Response Percent
Yes						85.70%
No						14.30%
Q2. Which best describes your job function						
Answer Options						Response Percent
Manager						16.70%
Project Manager						83.30%
Choose not to Answer						0.00%
Q3. Does the DBIDS Master Plan Spreadsheet help you in managing your job?						
Answer Options						Response Percent
Definitely						33.30%
Probably						50.00%
Not sure						0.00%
Probably not						16.70%
Definitely not						0.00%
Q4. Can you get relevant data from the DBIDS Master Plan Spreadsheet easily?						
Answer Options						Response Percent
Definitely						16.70%
Probably						16.70%
Not sure						0.00%
Probably not						50.00%
Definitely Not						16.70%
Q5. When using DBIDS Master Plan Spreadsheet, how important were each of the following?						
Answer Options	Very Unimportant	Unimportant	Important	Very Important	N/A	Rating Average
Functionality	1	0	1	4	0	3.33
Accessibility	1	0	1	4	0	3.33
Security	1	0	5	0	0	2.67
Q6. How would you rate your satisfaction with DBIDS Master Plan Spreadsheet?						
Answer Options	Very Dissatisfied	Dissatisfied	Satisfied	Very Satisfied	N/A	Rating Average
Functionality	1	4	1	0	0	2
Accessibility	1	3	1	1	0	2.33
Security	0	2	3	1	0	2.83
Q7. Overall, the DBIDS Master Plan Spreadsheet allows you to complete your job efficiently and effectively?						
Answer Options						Response Percent
Strongly Agree						0.00%
Agree						0.00%
Neutral						16.70%
Disagree						83.30%
Strongly Disagree						0.00%

<b>Q8. What about DBIDS Master Plan Spreadsheet causes you the most frustration?</b>												
I mostly access the Quick Stats worksheet at the beginning when responding to questions from DMDC personnel - leadership or those working on projects for leadership. I cannot trust the numbers as they are always incorrect for one reason or another.												
Clunky; hard to read, use & update; oversized; no ability to print a report or selected data; very, very slow to work with.												
The large amount of data in each row; the inability to associate multiple action items with a site; the lack of historical data on equipment orders, installs, and estimates for life-cycle replacements.												
Inputting data only to have to recreate spreadsheets, reports and summaries on my own because I cannot pull them out of the spreadsheet easily.												
Difficult to update, unwieldy to view												
<b>Q9. Based on your experience, would you recommend DBIDS Master Plan Spreadsheet to a co-worker?</b>												
<table border="1"> <thead> <tr> <th>Answer Options</th><th>Response Percent</th></tr> </thead> <tbody> <tr> <td>Definitely</td><td>0.00%</td></tr> <tr> <td>Probably</td><td>16.70%</td></tr> <tr> <td>Not sure</td><td>16.70%</td></tr> <tr> <td>Probably not</td><td>50.00%</td></tr> <tr> <td>Definitely not</td><td>16.70%</td></tr> </tbody> </table>	Answer Options	Response Percent	Definitely	0.00%	Probably	16.70%	Not sure	16.70%	Probably not	50.00%	Definitely not	16.70%
Answer Options	Response Percent											
Definitely	0.00%											
Probably	16.70%											
Not sure	16.70%											
Probably not	50.00%											
Definitely not	16.70%											
<b>Q10. What specific suggestions do you have that will help us improve DBIDS Master Plan Spreadsheet?</b>												
1). You didn't list accuracy as a desired feature - perhaps that's part of functionality. Need to be able to quickly see how many DBIDS sites are installed in each region, by country and/or city/state. It would be most useful to be able to click on country/state and get a listing of which sites are included. Now I have to go to individual worksheets to find, for example, which sites are included in the count for Florida. (And the count is currently wrong.) Need to be able to generate own reports and have the ability to select desired fields. Finally, if I can't get to the database or it's on a platform not approved by DMDC it won't do me any good. I'm located in Arlington, VA and I don't know if Access 2007 is available to us.												
2). Needs to be user friendly - easy to update; configurable views/sorts for updating.												
3). Get rid of the spreadsheet; use Project or some other application that manages the data more effectively.												
4). Making updates easier for all rather than having to be made to a single spreadsheet and making it easier to pull canned reports or do ad hoc queries of the data being stored in it.												
5). By site update capability with all appropriate fields; better rollup of statistics												
6). Consider it is just a single Excel spreadsheet that many people access and edit. I don't like that changes I make to my sections overwrite the entire document i.e. overwrite other managers work. I don't like that I can only access it from my OCONUS location must be done through a VPN to the DMDC network, thus making my work to the current document extremely slow. Scrolling across a line item in this huge spreadsheet is extremely frustrating. It would be better to see more of the information on screen such as on a form.												

DBIDS Master Plan Database Employee (To-Be) Questionnaire					
Q1. Do you use the DBIDS Master Plan Database?					
Answer Options					Response Percent
Yes					100.00%
No					0.00%
Q2. Which best describes your job function					
Answer Options					Response Percent
Manager					14.30%
Project Manager					71.40%
Choose Not To Answer					14.30%
Q3. Does the DBIDS Master Plan Database help you in managing your job?					
Answer Options					Response Percent
Definitely					57.10%
Probably					42.90%
Not Sure					0.00%
Probably Not					0.00%
Definitely Not					0.00%
Q4. Can you get relevant data from the DBIDS Master Plan Database easily compared to the last system?					
Answer Options					Response Percent
Definitely					85.70%
Probably					14.30%
Not Sure					0.00%
Probably Not					0.00%
Definitely Not					0.00%
Q5. When using DBIDS Master Plan Database, how important were each of the following?					
Answer Options		Very Unimportant	Unimportant	Important	Very Important
Functionality		0	0	1	6
Accessibility		0	0	1	6
Security		0	0	4	3
Q6. How would you rate your satisfaction with DBIDS Master Plan Database?					
Answer Options		Very Dissatisfied	Dissatisfied	Satisfied	Very Satisfied
Functionality		0	0	3	4
Accessibility		0	0	3	4
Security		0	0	4	3
Accuracy		0	0	3	3
Q7. Overall, does the DBIDS Master Plan Database allow you to complete your job efficiently?					
Answer Options					Response Percent
Strongly Agree					42.90%
Agree					57.10%
Neutral					0.00%
Disagree					0.00%
Strongly Disagree					0.00%
Q8. What about the DBIDS Master Plan Database causes you the most frustration?					
No responses					

<b>Q9. Based on your experience, would you recommend the DBIDS Master Plan Database to a co-worker?</b>	
<b>Answer Options</b>	<b>Response Percent</b>
<b>Definitely</b>	42.90%
<b>Probably</b>	57.10%
<b>Not Sure</b>	0.00%
<b>Probably Not</b>	0.00%
<b>Definitely Not</b>	0.00%

<b>Q10. What specific suggestions do you have that will help us improve DBIDS Master Plan Database?</b>
1). Ability to cross reference with authoritative dbase to prevent duplication of effort and/or repeat entries
2). I continue to receive errors and get kicked out of the app - can usually go back in easily and seems to work better if I search with a site id vs selecting from the drop down. We also have some additional workstation types that could be added rather than having to track in the remarks area.
3). The overall rollup sheet is the one I use most...now it's very small and hard to read...but it does give me what I need!

<b>Q11. Would you say the deployment of DBIDS Master Plan Database system has changed your business process? If so in what ways?</b>
1). Yes, it has improved the way I spend my time while VPN'd in to access and use the Master Plan.
2). potential exists...
3). I haven't had an opportunity to utilize the reporting capability but know this will be a great aid to Managers who need to look at all service data collectively.
4). Not really.
5). It has made it easier for the Project Managers to update info, making it more likely that it will actually get updated!!! Contributing, of course, to a better database...

## APPENDIX B—AS-IS INTERVIEWS

### Question 1

**LT Clarke:** What is essentially DBIDS?

**Stakeholder:** A defense department design project for access control for all military bases. Its usage is predominately focused on those folks who do not have a CAC card per say. It is meant more for the vendors that come out on repetitive bases to control their access to the base over a period of time. The system does take CAC cards also for retirees.

### Question 2

**LT Clarke:** Who updates the DBIDS Master Plan Spreadsheet System?

**Stakeholder:** The original spreadsheet I was given the responsibility for making sure that it was ok and operational. The PM (project managers) for the regions were required for their job performance to enter and keep their data accurate and up to date; and actually they were suppose to enter it twice a month.

### Question 3

**LT Clarke:** So they were actually required to enter the data twice a month?

**Stakeholder:** Yes, it was a business requirement for their objectives on a yearly basis.

### Question 4

**LT Clarke:** Could you describe the current business process for the DBIDS Master Plan Spreadsheet system?

**Stakeholder:** Well the spreadsheet was kept on a share drive that was accessible by all the PMs and the management team. It was a three peace structure me being responsible, PMs keeping the data accurate and up to date and the managers being able to look at this excel spreadsheet would theoretically have some top level picture I guess of their business as a whole of projects.

### Question 5

**LT Clarke:** So as I understand does the manager enters the database the manager which is yourself would you agree that the process flow starts at the managers then to you the Operations Program Manager. For instance, you are requested by the manager to provide certain data and you will provide that date expeditiously.

**Stakeholder:** Right it is fairly accurate.

**Question 6**

**LT Clarke:** When did you realize there was a problem with the current system?

**Stakeholder:** Well when the first time when somebody put spaces or zeros in top of the worksheet did not roll up, I had to go find where the space was because the number did not work correctly. That made it pretty evident that there was a problem.

**Question 7**

**LT Clarke:** So the burden was in regards to the business process you were sort of the bottleneck. You had to spend allot of time correct in accuracies.

**Stakeholder:** The first time had to go through and fix excel it took me the better part of a day actually going to find where all those spaces and zeros were and also where the formulas where screwed up.

**Question 8**

**LT Clarke:** Was the As-is business process effective?

**Stakeholder:** No it was not for allot of reasons.

**Question 9**

**LT Clarke:** What I meant did the As-is business process do its job?

**Stakeholder:** To a degree however, it was an afterthought for most PMs because everyone depended there business on the numbers that were in the DBIDS Master Plan Spreadsheet system and they were not accurate and I knew they weren't accurate. They have not been updating for a while.

**Question 10**

**LT Clarke:** Was there a process to select the As-is system?

**Stakeholder:** Yea one of the PMs was ask to sit down with a group and ask what is it that we want to find out. So they started building an excel spreadsheet in column to column and inputted all the information. The excel spreadsheet kept growing and growing and growing. Unfortunately not allot of thought was put into the development because the system. No question about what is the data we really mean what is the measurable business metric that we want to look for. Yea it would be nice we need to know about BFAS (some type of equipment information irrelevant to the interview). So they started building columns but the problem with that is you grow from the first dozen columns then you go out to 40+ it becomes untraceable when navigating the spreadsheet. You would not know for sure if they were actually inputting the right data in the right row. Now with data is visually in one place not having to search for rows. The can query a sub menu and look for the information that they want or to update. Or if they are more technical have the spreadsheet view incorporated with the database.

**Question 12**

**LT Clarke:** Who are performers?

**Stakeholder:** Well it was another operations project manager (name omitted) and I. We are the operations leads managers for the project. The local manager (name omitted) is the operations boss out here (California). The PMs has changed over time but, predominately the PMs in the Air force site PM and a PM assigned to South West Asia. He is the one who requested your assistance in this matter from NPS. The Enterprise Operations Project Manager is located on the east coast.

**Question 13**

**LT Clarke:** Was there a Feasibility Study done prior to using the spreadsheet or did we look at other options.

**Stakeholder:** No spreadsheets are what they are comfortable with here.

**LT Clarke:** But did they address what other options could be used.

**Stakeholder:** No, it cried out to put into database because of the share level of information to be gathered.

**Question 14**

**LT Clarke:** Do you remember

**Stakeholder:** No, I have to go back and look at when we came over to NPS with our issues.

**Question 15**

**LT Clarke:** Security features; Were there an issue with DBIDS Master Plan Spreadsheet System?

**Stakeholder:** there weren't any ... And that was largely the problem because there was no tracking no tracking for security we did not know when anyone entered the spreadsheet to make changes. Never could track when or who made changes. It was a real pain when requested for the need to tell when who updated the spreadsheet. The PMs don't report to me I mean I can ask them to tell me but if they don't tell me I have no mechanism to see when or if they do. You see the way that I can see when or if they do is when I can get into a database. In a database I can track when logging has been done you can know who has been making changes.

**Question 16**

**LT Clarke:** What are your major bottleneck with the current system where you said you have to do all the checks and updates?

**Stakeholder:** Right that's the major bottlenecks. Obviously as a flat file, it is easier from a performance point of view from a system level performance point of view. Being a database it becomes a little more complicated a little more depth that is needed. That is why we kind of had to spread out and go to the spread sheet version for the performance. Performance was painful

to go through the menu. For instance, today I had to be on the phone with a regional PM that they could not receive or send data. Their network pipes (connectivity) really has slow throughput.

**Question 17**

**LT Clarke:** So you had to do a lot of rework with the As-is?

**Stakeholder:** I had no idea when they were doing updates to the As-is system so I would always ask is it accurate and I would always compare it against other entities.

**Question 18**

**LT Clarke:** So you had to take their version of the spreadsheet and compare it to yours?

**Stakeholder:** Yes, for instance, a vendor indicates there is 6 systems at a site and when I compare it to spreadsheet that is supposedly accurate it tells me that there is only 3. To this day, we are still doing this since we are in our maintenance update phase.

**Question 19**

**LT Clarke:** What were your method of communication?

**Stakeholder:** Email

**Question 20**

**LT Clarke:** Do you send entire file for instance 2meg?

**Stakeholder:** No we simply state in an email that they had completed the update.

**Question 21**

**LT Clarke:** So there was no data transfer limitation?

**Stakeholder:** Yes

**Question 22**

**LT Clarke:** So was the data sent via share point?

**Stakeholder:** never got to be placed on a SharePoint server but it was on a share drive. We would have liked it to be on a SharePoint.

**Question 23**

**LT Clarke:** What skill sets did program managers have?

**Stakeholder:** Most were PMs skilled on excel it wasn't that complicated.

**Question 24**

**LT Clarke:** Is this system a front end or back end system

**Stakeholder:** currently it is a wow that's a good question

**LT Clarke:** We can come back to that question.

**Stakeholder:** I would say it is more a front-end system but from an enterprise point of view, it is more of a backend system.

**Question 25**

**LT Clarke:** Why wasn't a system that could do all this thought about or considered?

**Stakeholder:** This was due partly to money partly the big picture view

**NOTE:** Questions and responses were transcribed as is and format might not meet the correct grammatical format.

**Question 1**

**LT Clarke:** What is your current business spreadsheet process when requesting reports for site updates?

**Stakeholder:** The process would start by me requesting updates via E-mail. This was labor intensive because, I had no consistency with the process.

**Question 2.**

**LT Clarke:** Was the process efficient and effective?

**Stakeholder:** No again, it was very time consuming.

**Question 3**

**LT Clarke:** Were you knowledgeable about sites status both CONUS and OCONUS to clearly manage the business metric?

**Stakeholder:** I felt I knew what was happening within the business simply because of my familiarity with the DBIDS Spreadsheet. However, I would be the single point of failure. If someone else were to take over management, they would have a hard time getting up to speed with what was going on.

**Question 4**

**LT Clarke:** Could you estimate the average waiting time from a requested report until it was received?

**Stakeholder:** Due to time zone issues, I have been accustomed to getting reports back within a 24 hour (day) period.

**LT Clarke:** Wow that is a long time.

**Question 5**

**LT Clarke:** Did you initiate the change to transition to a better system?

**Stakeholder:** I was complaining about the issues but that trigger was pulled by Mike Milos. So no.

**LT Clarke:** Thank you for your time.

THIS PAGE INTENTIONALLY LEFT BLANK

## APPENDIX C—TO-BE INTERVIEWS

### Question 1

**LT Clarke:** Do you agree that the later To-be system is better in regards to speed usability compared to the spreadsheet. The security concerns that an Access Database.

**Stakeholder:** oh night and day the changes has been huge; the ability to go in and take a snapshot view of individual sites to do the quick state report with the added capability to drill down and view individual COCOMs (Combatant Commanders) command level to state a country whatever it make a huge impact. For instance, before you had to set up report structures and design a view in excel now you can spit out exactly a quick stat report of whatever we select. So it is a huge difference in regards to performance with the ability to give at least that to date a snapshot matrix for the business.

### Question 2

**LT Clarke:** Does it address accuracy?

**Stakeholder:** Yes, in fact it does accuracy in excel has a propensity if you put zero or commas or wrong number or put spaces in the stats does not roll up. Accuracy at the worksheet level that's not an issue anymore because they are forced to actually enter the information into the database ( Microsoft Access 2003 database ) and in the correct way because a template has been established only enter it one way. Some of the examples would be like dates where before PMs would put in fiscal 09 or fiscal 010 now they are actually required to put in one single standard date.

### Question 3

**LT Clarke:** What was the reason for the technology change? Or was it business driven?

**Stakeholder:** From my perspective, it was a technology change to facilitate better knowledge roll up of the business metric.

### Question 4

**LT Clarke:** Was this a hybrid of two things

**Stakeholder:** Yes it was

### Question 5

**LT Clarke:** Does the To-be offer more security than the spreadsheet.

**Stakeholder:** Absolutely

**Question 6**

**LT Clarke:** What would be the next stage of evolution for the To-be system?

**Stakeholder:** Yes, this is a short-term solution to fill the current issues by we have intentions on introducing this system to the IT. It is out on the share drive right now but I have been working with the management team that this Database needs to be transferred from the share drive to be owned by the IT. During testing going to leave it were it is.

**Question 7**

**LT Clarke:** Added some more reports to the To-be system in what way does that help?

**Stakeholder:** It brings to bear more management insight to the information and it was requested when the enterprise PM conducted walkthrough of the process. The program manager wanted to know why we are still using spreadsheet and that it needed to be in a database. Great question I replied. There is a place we just need to design and implement it. Your partner and I wanted to see how fast we could turn this around because it is beneficial from a business point of view. I kind of expanded it to the installations also and tracking finding out how long it takes to do an installation because, this metric I believe is really important due to cost incurred. We really need to drive towards making that process more streamlined.

**LT Clarke:** Is your business process for request of report for sit updates the same with Database system? If not what changed?

**Stakeholder:** No, I just go in and get stuff. The system now is more accurate and PM are more cognizant about keeping updates current. But, data is only as good as what is inputted. Garbage in Garbage out.

**Question 2**

**LT Clarke:** Is the process more efficient and effective?

**Stakeholder:** Absolutely

**Question 3**

**LT Clarke:** Was more knowledge gained about the business metric?

**Stakeholder:** More knowledge; yes, because as the system was being tested we noticed that information in this form will be valuable. So, we were requesting new reports that would cover the business view of DBIDS platforms being fielded. For instance, Southwest Asia PM did not want to open numerous windows to update a single number. Instead, we allowed for a spreadsheet view that he could go down the rows of that particular spreadsheet column and update a lot faster.

**Question 4**

**LT Clarke:** Could you estimate the average waiting time from a requested report until it was received?

**Stakeholder:** This depends on where the project manager how good they are at keeping things updated. Realistically less than 12-hour time frame

**Question 5**

**LT Clarke:** Do you think the To-be system has the capability to develop into a more robust database, linking supply and other IS technology for cross functions?

**Stakeholder:** Sure, for instance in or meeting today about the continuing life cycle of To-be system Documents could be entered as PDF. This will enable to project manager to have more of a one-stop shop for information to conduct their work.

**Question 6**

**LT Clarke:** This is not on my interview but personally do you feel that now your able to do more work concerned with the DBIDS?

**Stakeholder:** I believe that there is more accountability with this system being fielding

**LT Clarke:** This concludes the interview thank you for your time for giving this opportunity to help

THIS PAGE INTENTIONALLY LEFT BLANK

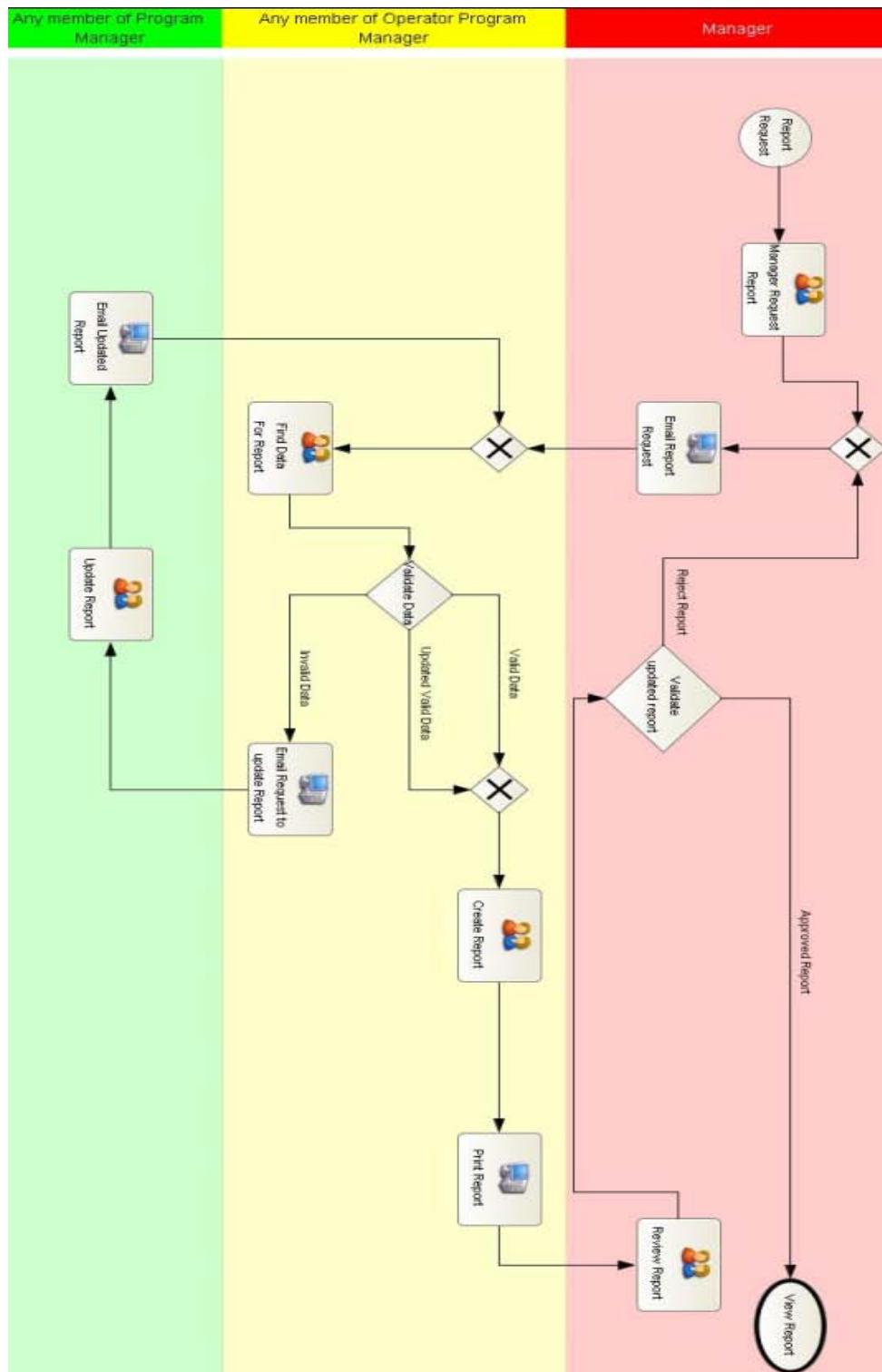
## APPENDIX D—PIECES FRAMEWORK CHECKLIST

A checklist for identifying problems with an existing information system.

- Performance
  - Throughput
  - Response Time
- Information (and Data)
  - Outputs
    - Lack of any information
    - Lack of necessary information
    - Lack of relevant information
    - Too much information – information overload
    - Information that is not in a useful format
    - Information that is not accurate
    - Information that is difficult to produce
    - Information that is not timely to its subsequent use
  - Inputs
    - Data is not captured
    - Data is not captured in time to be useful
    - Data is not accurately captured – contains errors
    - Data is difficult to capture
    - Data is captured redundantly – same data is captured more than once
    - Too much data is captured
    - Illegal data is captured
  - Stored Data
    - Data is stored redundantly in multiple files and/or databases
    - Stored data is not accurate
    - Data is not secure from accident or vandalism
    - Data is not well organized
    - Data is not flexible – not easy to meet new information needs from stored data
    - Data is not accessible
- Economics
  - Costs
    - Costs are unknown
    - Costs are untraceable
    - Costs are too high
  - Profits
    - New markets can be explored
    - Current marketing can be improved
- Control (and Security)
  - Too little security or control

- Input data is not adequately edited
  - Crimes (e.g. fraud, embezzlement) are (or can be) committed against the data
  - Ethics are breached on data or information – refers to data or information getting to unauthorized people
  - Redundantly stored data is inconsistent in different files or databases
  - Data privacy regulations or guidelines are being (or can be) violated
  - Processing errors are occurring (either by people, machines, or software)
  - Decision-making errors are occurring
- Too much control or security
  - Bureaucratic red tape slows the system
  - Controls inconvenience customers or employees
  - Excessive controls cause processing delays
- Efficiency
  - People, machines, or computers waste time
    - Data is redundantly input or copied
    - Data is redundantly processed
    - Information is redundantly generated
  - People, machines, or computers waste materials and suppliers
    - Effort required for tasks is excessive
    - Materials required for tasks is excessive
- Service
  - The system produces inaccurate results
  - The system produces inconsistent results
  - The system produces unreliable results
  - The system is not easy to learn
  - The system is not easy to use
  - The system is awkward to use
  - The system is inflexible to new or exceptional situations
  - The system is inflexible to change
  - The system is incompatible with other systems
  - The system is not coordinated with other systems
-

## APPENDIX E—PROCESS SIMULATIONS



As-Is Simulation Results					
Duration	123:48:00	Time			

Process Time And Cost					
Process	Scenario	Instances	Total Cost	Waiting Time (Time)	Total Time (Time)
Process	DBIDS Spreadsheet	100	\$17451.84	6540:32:00	7031:46:00

Process	
Scenario	DBIDS Spreadsheet
Instances	100

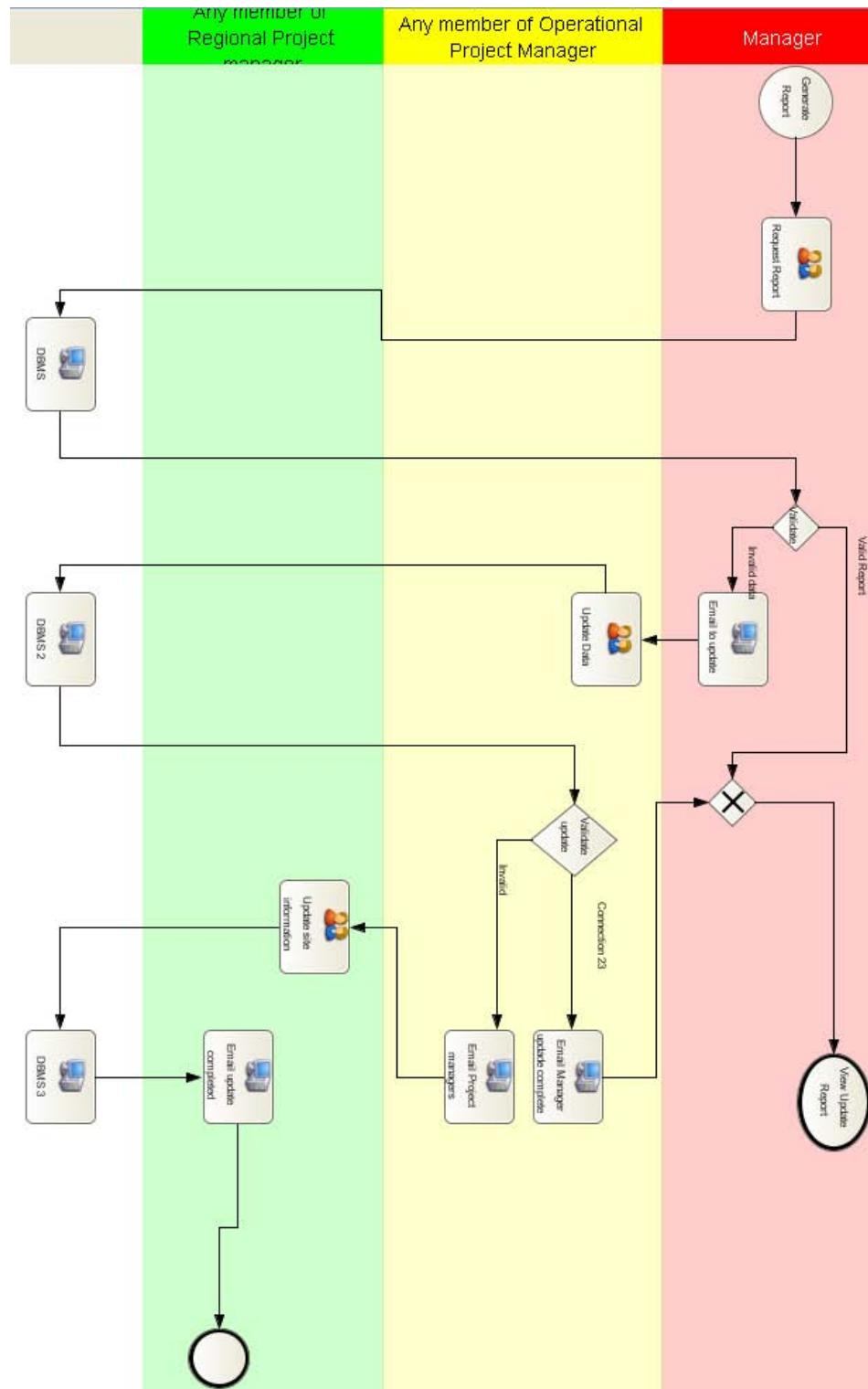
Activity	Performer	Occurs	Waiting Time (Time)	Time To Complete (Time)	Total Time (Time)
Create Report	Any member of Operator Program Manager	111	2358:08:00	18:30:00	2376:38:00
Find Data For Report	Any member of Operator Program Manager	202	4162:08:00	214:33:00	4376:41:00
Manager Request Report	Manager	100	0:10:00	3:20:00	3:30:00
Review Report	Manager	111	0:00:00	9:15:00	9:15:00
Update Report	Any member of Program Manager	91	0:00:00	187:31:00	187:31:00
Email Report Request	Generic	111	8:31:00	9:15:00	17:46:00
Email Request to Update Report	Generic	91	3:17:00	15:10:00	18:27:00
Email Update Report	Generic	91	2:49:00	15:10:00	17:59:00
Email Updated Report	Generic	111	5:29:00	18:30:00	23:59:00

Resource	Unit	Cost/ Unit	Threshold	Usage	Cost
Manager	Hour	50	0	12	\$600.00
Any member of Operator Program Manager	Hour	40	0	233	\$9320.00
Any member of Program Manager	Hour	40	0	187	\$7480.00
Generic	Invocation	0	0	404	\$0.00

Performers Queue Length and Utilization						
Name	Average	Min	Max	Utilized(%)	Idle(%)	
Manager	0	0	1	10.16	89.84	
Any member of Operator Program Manager	52.67	0	93	94.12	5.88	
Value of 'Creator'	0	0	0	0	100	
Any member of Program Manager	0	0	0	25.24	74.76	
Generic	0.16	0	4	46.92	53.08	

Bottlenecks					
Process	Activity	Performer	Avg Queue Length	Min Queue Length	Max Queue Length
Process	Create Report	Any member of Operator Program Manager	19.05	0	44
Process	Find Data For Report	Any member of Operator Program Manager	33.62	0	79
Process	Manager Request Report	Manager	0	0	1
Process	Review Report	Manager	0.07	0	4
Process	Email Report Request	Generic	0.03	0	1
Process	Email Request to Update Report	Generic	0.02	0	1
Process	Email Update Report	Generic	0.04	0	1
Process	Email Updated Report	Generic	19.05	0	44

<b>Note:</b>	Red-marked Waiting Time values indicates "Activity has waiting time"
	Red-marked Usage values indicates "Usage crossed threshold"



To-Be Simulation Results					
Duration	107:51:00	Time			

Process Time And Cost					
Process	Scenario	Instances	Total Cost	Waiting Time (Time)	Total Time (Time)
Process	To-Be Database	100	273.33	0:44:00	11:49:00

Process	
Scenario	DBIDS Spreadsheet
Instances	100

Activity	Performer	Occurs	Waiting Time (Time)	Time To Complete (Time)	Total Time (Time)
Request Report	Manager	100	0:02:00	3:20:00	3:22:00
Update Data	Any member of Operational Project Manager	5	0:00:00	2:35:00	2:35:00
Update site information	Any member of Regional Project Manager	1	0:00:00	0:05:00	0:05:00
Access Database	Generic	100	0:39:00	3:20:00	3:59:00
Access Database_2	Generic	1	0:00:00	0:00:00	0:00:00
Email Manager update complete	Generic	4	0:00:00	0:40:00	0:40:00
Email Project Managers	Generic	1	0:00:00	0:10:00	0:10:00
Email to update	Generic	5	0:02:00	0:50:00	0:52:00
Unrestricted Access Database	Generic	5	0:01:00	0:05:00	0:06:00

Resource	Unit	Cost/Unit	Threshold	Usage	Cost
Manager	Hour	50	0	3	\$150.00
Any member of Operational Project Manager	Hour	40	0	2	\$80.0
Any member of Regional Project Manager	Hour	40	0	0	\$0.00
Generic	Invocation	0	0	116	\$0.00

Performers Queue Length and Utilization					
Name	Average	Min	Max	Utilized(%)	Idle(%)
Manager	0	0	1	18.57	81.43
Any member of Operator Program Manager	0	0	0	7.2	92.8
Value of 'Creator'	0	0	0	0	100
Any member of Program Manager	0	0	0	0.09	99.91
Generic	0.04	0	1	28.32	71.68

Bottlenecks					
Process	Activity	Performer	Avg Queue Length	Min Queue Length	Max Queue Length
To_be_Model_V1	Request Report	Manager	0	0	1
To_be_Model_V1	Access Database	Generic	0.04	0	1
To_be_Model_V1	Email to update	Generic	0	0	1
To_be_Model_V1	Unrestricted Access Database	Generic	0	0	1

<b>Note:</b>	Red-marked Waiting Time values indicates "Activity has waiting time"
	Red-marked Usage values indicates "Usage crossed threshold"

## APPENDIX F—STAKE HOLDER GLOSSARY

Stakeholder	Description	Use-Case
<b>DBIDS Managers</b>	A user with viewing privileges to the database.	View Site Info Generate Quick Stats Report Generate Maint Report Generate Pending Site Report Generate Contact Info Report
<b>DBIDS Administrators (Operations PMs)</b>	A user with full control over the database.	Add/Delete/View/Edit Site & Pending Site Info Restore Deleted Sites Add/Edit Action Item Add/Edit Contact Info View/Clear Audit Trail Generate Quick Stats Report Generate Maint Report Generate Pending Site Report Generate Site Survey Report Generate Contact Info Report View/Edit COCOM Info View/Edit Service Branch Info View/Edit Location Info View/Edit Project Manager Info View Planned & Installed Errors View/Edit Project Manager Filters
<b>Regional PMs</b>	A user with responsibility over a COCOM and or Service Branch. These users will have specific knowledge about each site under their responsibility. They will update information dealing with a site at a minimum of weekly. The most accessed information is the planned & installed values and current maintenance required at a specific site.	Add/Delete/View/Edit Site & Pending Site Info Generate Quick Stats Report Generate Maint Report Generate Pending Site Report Generate Contact Info Report

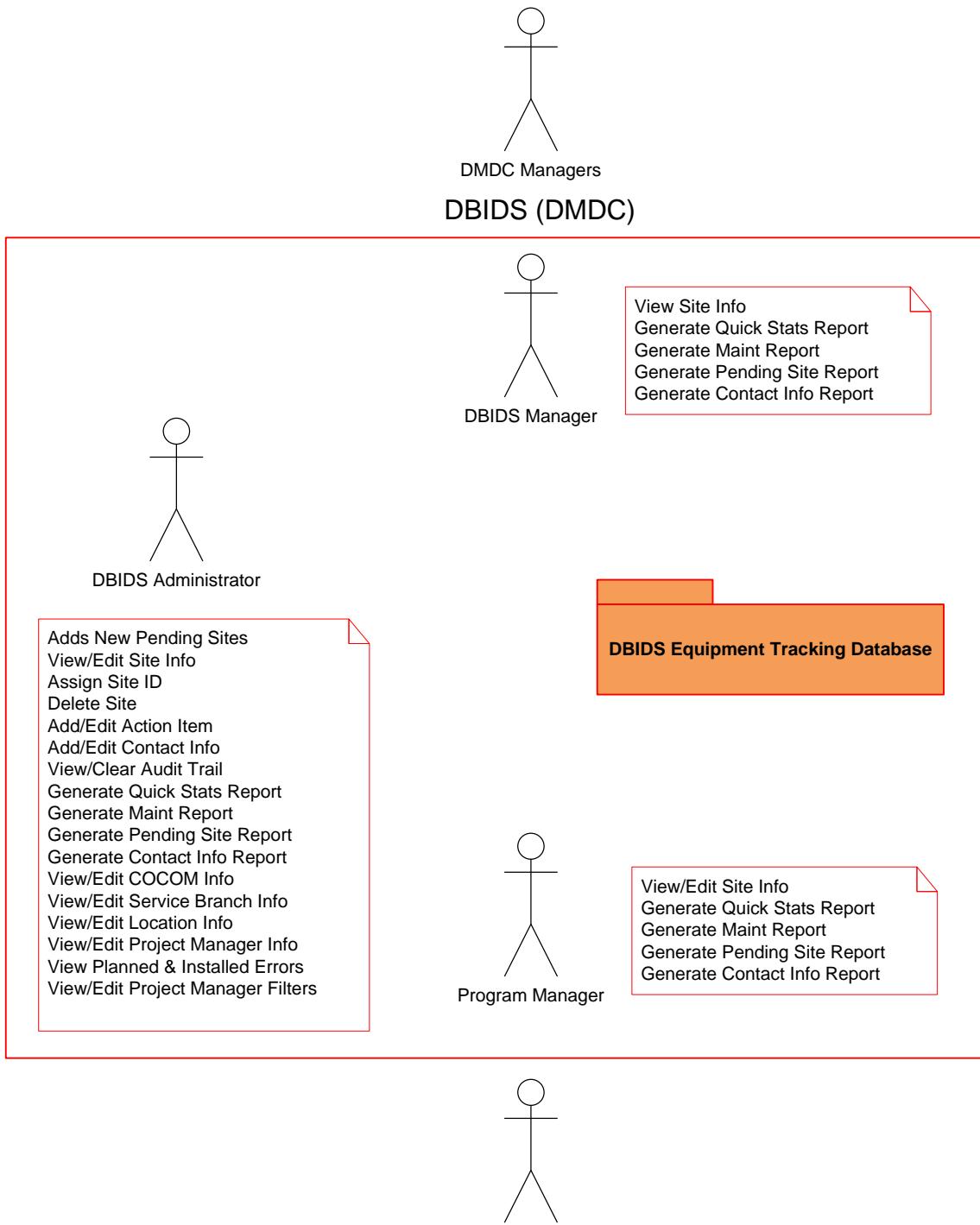
THIS PAGE INTENTIONALLY LEFT BLANK

## APPENDIX G—USE CASE GLOSSARY

Use-Case Name	Use-Case Description	Participating Actors & Roles
<b>Add/Delete/View/Edit Site &amp; Pending Site Info</b>	Allows users to add, delete, view, or edit site specific data and related records.	DBIDS Administrator DBIDS Manager (View Only) Regional PMs (Responsible Sites Only)
<b>Add/Edit Action Item</b>	Administrators can track specific “in house” action items and associate documentation and events to one or more sites.	DBIDS Administrator
<b>Add/Edit Contact Information</b>	Administrators can maintain PPI information of points of contact (POC) for each site. A POC may be assigned to more than one site.	DBIDS Administrator
<b>View/Clear Audit Trail</b>	Administrators can audit changes made to site information. They may view specific changes made to each site. Additionally they may print a report to allow for easier viewing and periodically clear the trail.	DBIDS Administrator
<b>Generate Maintenance Report</b>	Filterable listing of all sites and their current Maintenance Status.	DBIDS Administrator DBIDS Manager Program Manger (Responsible Sites Only)
<b>Generate Quick Stats Report</b>	Filterable listing of Sites sorted and consolidated by COCOM, then Country/State, then by Service Branch. With the totaled values of site information at each level.	DBIDS Administrator DBIDS Manager Program Manger (Responsible Sites Only)
<b>Generate Pending Site Report</b>	Filterable listing of all Pending Sites and their associated Site Survey and Funding information.	DBIDS Administrator DBIDS Manager Program Manger (Responsible Sites Only)
<b>Generate Contact Information Report</b>	Filterable listing of all Contact Information by either Site or by Contact.  By Site: Sorted by Combatant Command, State/Country, Site, and Contact Billet.  By Contact: Sorted by Contact Name, COCOM, State/Country, Service Branch, Site Name, and Contact Billet.	DBIDS Administrator DBIDS Manager (By Site Only) Program Manger (By Site Only and Responsible Sites Only)
<b>Generate Site Survey Report</b>	Filterable listing of all tagged Site Survey Information.	DBIDS Administrator

Use-Case Name	Use-Case Description	Participating Actors & Roles
<b>Add/Delete/View/ Edit COCOM Information</b>	Administrators can add/edit/delete Combatant Commands.	DBIDS Administrator
<b>Add/Delete/View/ Edit Service Branch Information</b>	Administrators can add/edit/delete Service Branches.	DBIDS Administrator
<b>Add/Delete/View/ Edit Location Information</b>	Administrators can add/edit/delete Locations.	DBIDS Administrator
<b>Add/Delete/View/ Edit Project Manager Information</b>	Administrators can add/edit/delete Project Managers.	DBIDS Administrator
<b>View Planned &amp; Installed Errors</b>	Administrators can identify inconsistencies in planned and installed equipment values for sites. Such as more installed items than planned at a site. Allows a direct link to the identified site to allow them to correct information.	DBIDS Administrator
<b>View/ Edit Project Manager Filters</b>	Administrators can adjust the filters associated to a PM. These filters make it so a PM can only view/edit their responsible sites.	DBIDS Administrator

## APPENDIX H—SOCIAL NETWORK



(Updated 16 July 2009)

THIS PAGE INTENTIONALLY LEFT BLANK

## **APPENDIX I—DBIDS MASTER DATABASE HELP FILE**



**DBIDS**  
**MASTER DATABASE**  
**HELP FILE**

## **I. INTRODUCTION**

This document is intended for use with the final version of the DBIDS Master Database Version 1.33 updated on 4 September 2009.

The documentation provided here, is meant to aid current Database Administrators in the processes of maintaining the database in its current state, with step-by-step explanations on maintenance and management procedures. This document is not a tool to instruct an end user in the working of Microsoft Access, but to explain to an Access user steps and process peculiar to this specific database.

Additionally, there are explicit definitions and relationships of the individual items contained within the database. With this information a developer with an understanding of relational databases, Access, the Structured Query Language (SQL) and Visual Basic might expand upon or update particular portions in order to maintain the database in a concurrent state in relation to the business processes and rules that may change over time.

THIS PAGE INTENTIONALLY LEFT BLANK

## II. GAINING ACCESS

### A. Groups

There are five total permission groups within the database. Two are default groups from Microsoft Access that must remain Admins and Users; however, they are not given any actual permission to access the database. The remaining three groups cover the areas needed.

- Full Permissions – Have full access to everything within the database.
- Read-Only Users – Have only read and execute privileges to all items in the database.
- Update Data Users – Have various privileges to each item in the database.

### B. Users

Currently there are ten different users. Here again two are default users from Microsoft Access that must remain Admin and User; however, here again they are not given any actual permission to access the database. The user “Developer” is needed by the Visual Basic coding in the database. Developer is used from within the code to access the user listing and groups to identify who the current user is and their level of access. A key item to adding a new user to the database is to insure that an equivalent entry is added to the table, tbIPMFilters. This can be done by using the ViewPMFilters form.

- Full Permissions
  - DBIDS Administrator
- Read-Only Users
  - DBIDS Manager
- Update Data Users
  - Air Force CONUS PM
  - Army CONUS PM

- CENTCOM PM
- EUCOM PM
- Navy CONUS PM
- PACOM PM

### III. DATABASE MAINTENANCE & MANAGEMENT

#### A. Maintenance

Currently two areas require periodic maintenance applied by an administrator.

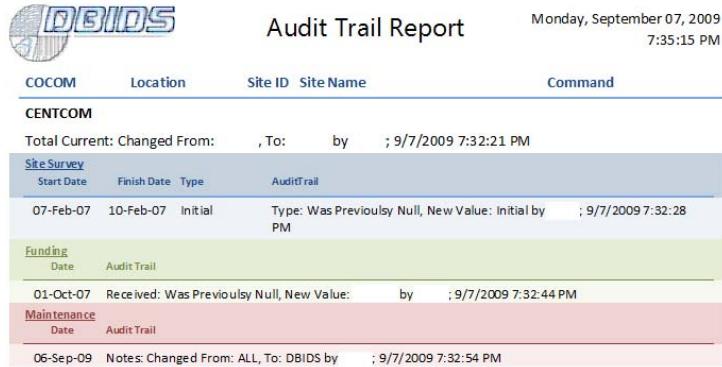
##### 1. Audit Trail

First, the Audit Trail needs to be cleared. To accomplish this, navigate using the built in switchboard to View Audit Trail Information. From this form (Figure 7), an administrator can view all changes to a Site's information made through the database's forms Sites, Pending Sites, Sites Datasheet, and Pending Sites Datasheet.

Figure 7. Audit Trail View Form

The administrator may then directly go to a site that is being viewed in the audit trail by selecting the “Goto Record” button. The periodic maintenance in this area is done in two steps.

- First, select the “Print” button, to print a report listing all of the current Audit trail entries (Figure 8).



The screenshot shows the DBIDS Audit Trail Report interface. At the top, there is a logo for DBIDS and the text "Audit Trail Report" followed by the date and time: "Monday, September 07, 2009 7:35:15 PM". The main area is a table with the following columns: COCOM, Location, Site ID, Site Name, and Command. The table has several rows, each representing an audit trail entry. The rows are color-coded: light blue for Site Survey, light green for Funding, light red for Maintenance, and light pink for Notes. The data in the table includes dates, times, and descriptions of the audit events.

COCOM	Location	Site ID	Site Name	Command
CENTCOM				
Total Current: Changed From: , To: by ; 9/7/2009 7:32:21 PM				
<u>Site Survey</u>				
Start Date	Finish Date	Type		Audit Trail
07-Feb-07	10-Feb-07	Initial		Type: Was Previously Null, New Value: Initial by ; 9/7/2009 7:32:28 PM
<u>Funding</u>				
Date				Audit Trail
01-Oct-07				Received: Was Previously Null, New Value: by ; 9/7/2009 7:32:44 PM
<u>Maintenance</u>				
Date				Audit Trail
06-Sep-09				Notes: Changed From: ALL, To: DBIDS by ; 9/7/2009 7:32:54 PM

Figure 8. Audit Trail Report

- Once all of the audit trail information has been printed, the administrator can then select the “Clear All Audit Trail Entries” button. By clearing the audit trail entries, it prevents the database from expanding too large.

## 2. Compacting & Repairing

The second area requiring periodic maintenance can be done anytime an administrator is in the database. Before exiting the database, the administrator need only select Tools → Database Utilities → Compact and Repair Database... (Figure 9). The database will then compress itself, keeping the overall size of the database low.



Figure 9. Compacting & Repairing

## B. Management

To conduct some changes to the database, such as adding new forms, an administrator must have the database opened exclusively. With users across the globe, this may not be an easy task to schedule. There is an ability to force users from the database.

### 1. Gaining Exclusive Access

To accomplish this task, an administrator must navigate, using the built in switchboard, to Open Database Manager. From this form, an administrator can view the current users accessing the database (Figure 10).

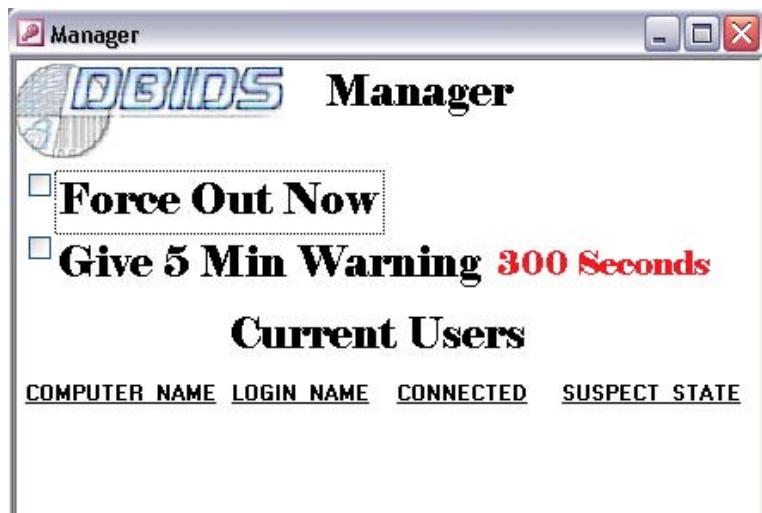


Figure 10. Database Manager Window

An administrator has two options. A five-minute warning, which will open a warning message on each of the user screens (Figure 11). After the countdown, all users will be removed from the database.



Figure 11. Exit Message

Alternatively, an administrator may immediately kick every user from the database. Once the administrator has removed all the users, they must reopen the database with the "Shift By-Pass." Then select File --> Open, navigating to the master file, and selecting the "Open Exclusive" button (Figure 12) insuring to use the "Shift By-Pass" during this opening.

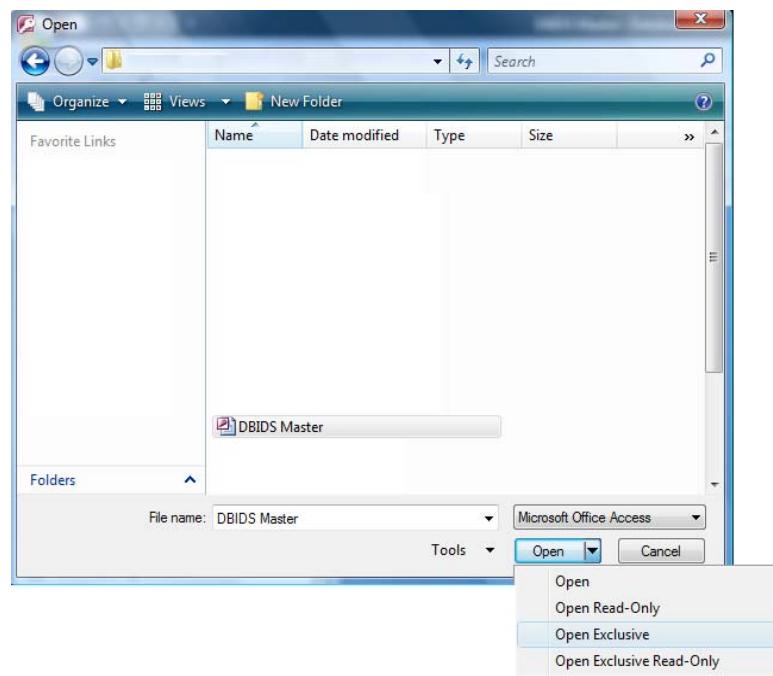
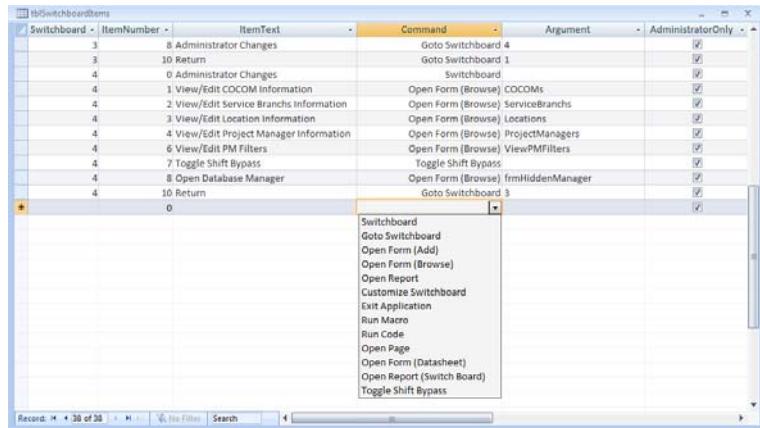


Figure 12. Open Exclusive

Now the administrator can accomplish the changes they require.

## 2. Adding to the Switchboard

Once a new item is created, the administrator may want to add the item to the built in switchboard. To accomplish this task requires a new entry in the table *tblSwitchboardItems* (Figure 13).



The screenshot shows a Microsoft Access table named 'tblSwitchboardItems'. The columns are: 'Switchboard', 'ItemNumber', 'ItemText', 'Command', 'Argument', and 'AdministratorOnly'. The table contains 10 rows of data. A new row is being added at the bottom, with the 'ItemText' field set to '0' and the 'Command' field set to 'Switchboard'. A context menu is open over this new row, listing various options such as 'Switchboard', 'Goto Switchboard', 'Open Form (Add)', etc.

Switchboard	ItemNumber	ItemText	Command	Argument	AdministratorOnly
	3	8 Administrator Changes	Goto Switchboard 4		
	3	10 Return	Goto Switchboard 1		
	4	0 Administrator Changes	Switchboard		
	4	1 View/Edit COCOM Information	Open Form (Browse)	COCOMr	
	4	2 View/Edit Service Branches Information	Open Form (Browse)	ServiceBranches	
	4	3 View/Edit Location Information	Open Form (Browse)	Locations	
	4	4 View/Edit Project Manager Information	Open Form (Browse)	ProjectManagers	
	4	6 View/Edit PM Filters	Open Form (Browse)	ViewPMFilters	
	4	7 Toggle Shift Bypass	Toggle Shift Bypass		
	4	8 Open Database Manager	Open Form (Browse)	frmHiddenManager	
	4	10 Return	Goto Switchboard 3		
	0		Switchboard Goto Switchboard Open Form (Add) Open Form (Browse) Open Report Customize Switchboard Exit Application Run Macro Run Code Open Page Open Form (Datasheet) Open Report (Switch Board) Toggle Shift Bypass		

Figure 13. Switchboard Items

## 3. Adding a New User

To add a new user to the database requires a current administrator to login and complete entries in two locations. The first location is within the Access Users Accounts, begin by selecting the “User and Group Accounts Button” on the main menu bar (Figure 14).

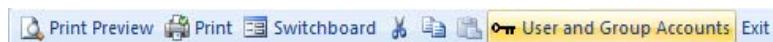


Figure 14. User and Group Accounts

On the Users Tab (Figure 15), click the “New” button and an additional popup window will appear, asking for a User Name and Personal ID. After entering the new user information, click “OK.”



Figure 15. Add New User

Once the new user has been entered, select the new user in the Name drop-down list (Figure 16).



Figure 16. Adding Group Memberships

Once the new user is selected, then add the appropriate Group Memberships, refer to the Gaining Access

Groups listed above. To do this, select the group you wish to add on the left then press the “Add>>” button. Now close the User and Groups Accounts popup. Using the built in switchboard, navigate to View PM Filters. Now, create a new user by clicking the new record button at the bottom of the form (Figure 17).



Figure 17. New Record Button

Enter the exact same name in the Current User field (Figure 18) as was used on the User Name field above (Figure 15).

A screenshot of a software application window titled "PM Filters". The window has a header with the "DBIDS" logo. Below the header, there are three input fields: "Current User:" with an empty text box, "COCOM Filter:" with an empty text box, and "Service Branch Filter:" with an empty text box. To the right of these fields is a "Apply Filter" button. Below these fields is a table with columns: "ID\_Site", "SiteName", "ServiceBranch", and "COCOM". The table is currently empty. At the bottom of the form, there is a "Site Count:" label followed by an empty text box.

Figure 18. Adding a new user to PM Filters

If this user is to have restrictions on which sites they may view, those filters must be entered at this time (Figure 19). Once a filter is entered, it can be tested here by pressing the “Apply Filter” button. The window in the middle of the form will show the sites that this user can view and show the count of the sites.

ID_Site	SiteName	ServiceBranch	COCOM
		Air Force	NORTHCOM
		Air Force	NORTHCOM
None		Air Force	NORTHCOM
		Air Force	NORTHCOM

Site Count: 8

Figure 19. Air Force NORTHCOM PM Filters

Lastly, after a user is added, that user should login into the database initially with a blank password. They may then immediately change their password to a private one at that time, by selecting the “User and Group Accounts” button on the main menu bar (Figure 14). Then selecting the “Change Logon Password” tab (Figure 20).

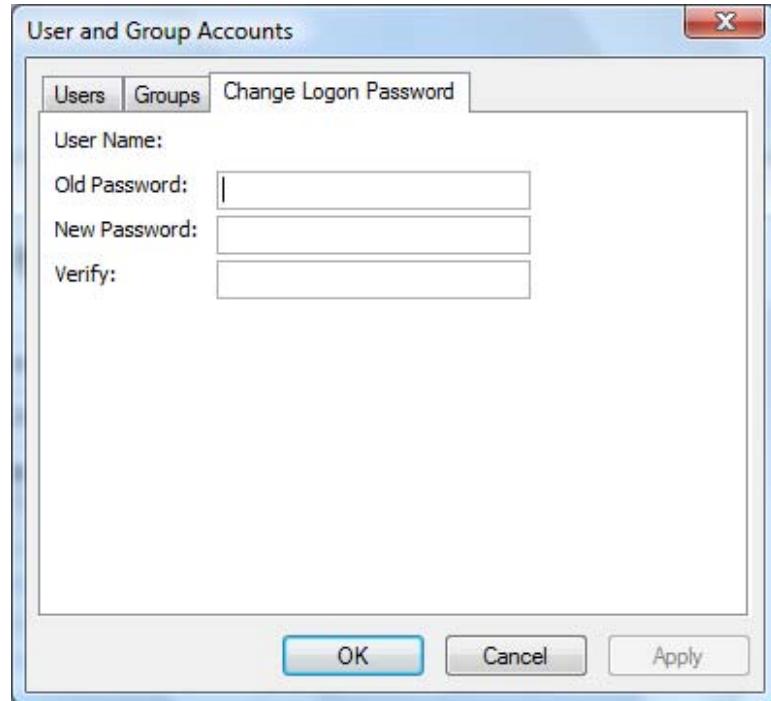


Figure 20. Change Logon Password

#### 4. Creating a Shortcut

To create a shortcut on a Windows based machine to open the database requires the following entry to be placed in the “Target” field of a shortcut (Figure 21).

*"Access Path" "DBIDS Master Path" /WRKGRP "DBIDS Access Path"*

- Access Path – The full path to the local installation of Microsoft Access Executable file (MSACCESS.EXE)
- DBIDS Master Path – The full path to the actual file DBIDS Master.mdb.
- DBIDS Access Path – The full path to the actual file DBIDS Access.mdw.

*"C:\Program Files\Microsoft Office\Office12\MSACCESS.EXE"*

*"C:\Users\Default\Desktop\DBIDS\DBIDS Master.mdb" /WRKGRP*

*"C:\Users\Default\Desktop\DBIDS\DBIDS Access.mdw"*

In the “Start in” field, insert the path to the directory where the DMDC Master database is located.

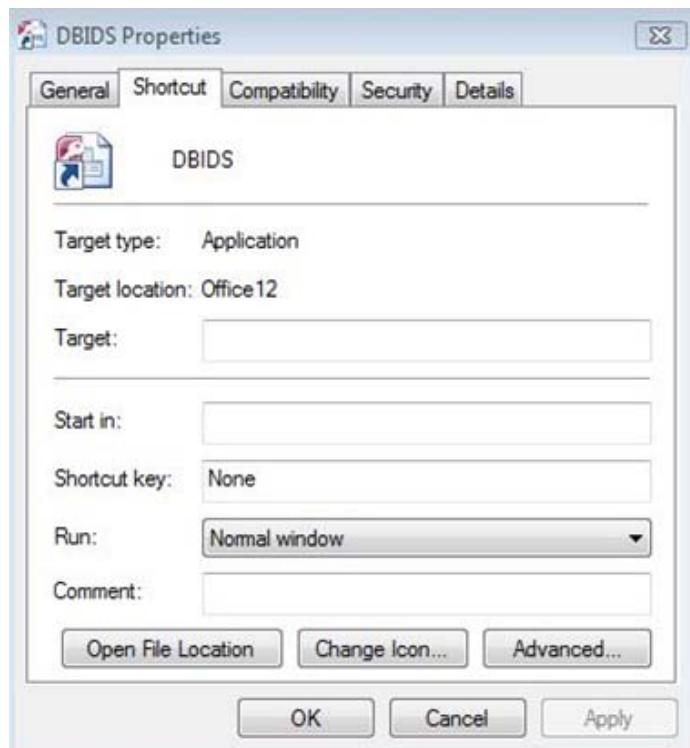


Figure 21. Shortcut

THIS PAGE INTENTIONALLY LEFT BLANK

## IV. SWITCHBOARD MAPPING



Item	Description	Target
1	View/Edit Site Information	Sites
2	Datasheet View Sites	SiteDatasheetView
3	View/Edit Pending Site Information	PendingSites
4	Datasheet View Pending Sites	PendingDatasheetView
5	Open Reports Filter (Dynamic Form depending on the Report being open, additionally limits filtering capabilities of Regional PMs to only sites within their responsibility.	frmOpenReports
6	View/Edit Contact Information (By Site)	ContactInformationBySite
7	View/Edit Contact Information (By Contact)	ContactInformationByContact
8	View/Edit Action Items	ActionItems
9	Shipping Address Datasheet View	SiteShippingAddress
10	View Audit Trail Information	AuditTrailView
11	Planned & Installed Errors	PlannedInstalledErrors
12	View/Restore Deleted Sites	SiteBackup
13	View/Edit COCOM Information	COCOMs
14	View/Edit Service Branch Information	ServiceBranchs
15	View/Edit Location Information	Locations
16	View/Edit Project Manager Information	ProjectManagers
17	View/Edit PM Filters	ViewPMFilters
18	Toggle Shift Bypass	Switchboard Private Function HandelButtonClick Select Case conCmdToggleBypass
19	Open Database Manager	frmHiddenManager
20	Quick Stats	rptQuickStats
21	Quick Stats (COCOMs Only)	rptQuickStatsCOCOM
22	Quick Stats (Audit View)	rptQuickStatsAudit
23	Maintenance Report	rptMaint
24	Pending Site Report	rptPendingSite
25	Contact Information Report (By Site)	rptContactInformationBySite
26	Contact Information Report (By Contact)	rptContactInformationByConact
27	Site Survey Report	rptSiteSurvey

	Only Viewable/Accessible by a DBIDS Administrator
--	---

## V. TABLES

The tables in the database are listed here. For each table the following information is listed:

- Table definition - A short explanation for inclusion and use in the database.
- User Permissions - Identifies the level of access the Update Data Users group have.
- Primary Key
- Foreign Keys
- Cascading update/delete relationships - Identifies if there are any linked tables that when a record is deleted from the main table it would remove any linked records from the linked tables.

To view table relationships refer to Figure 22.

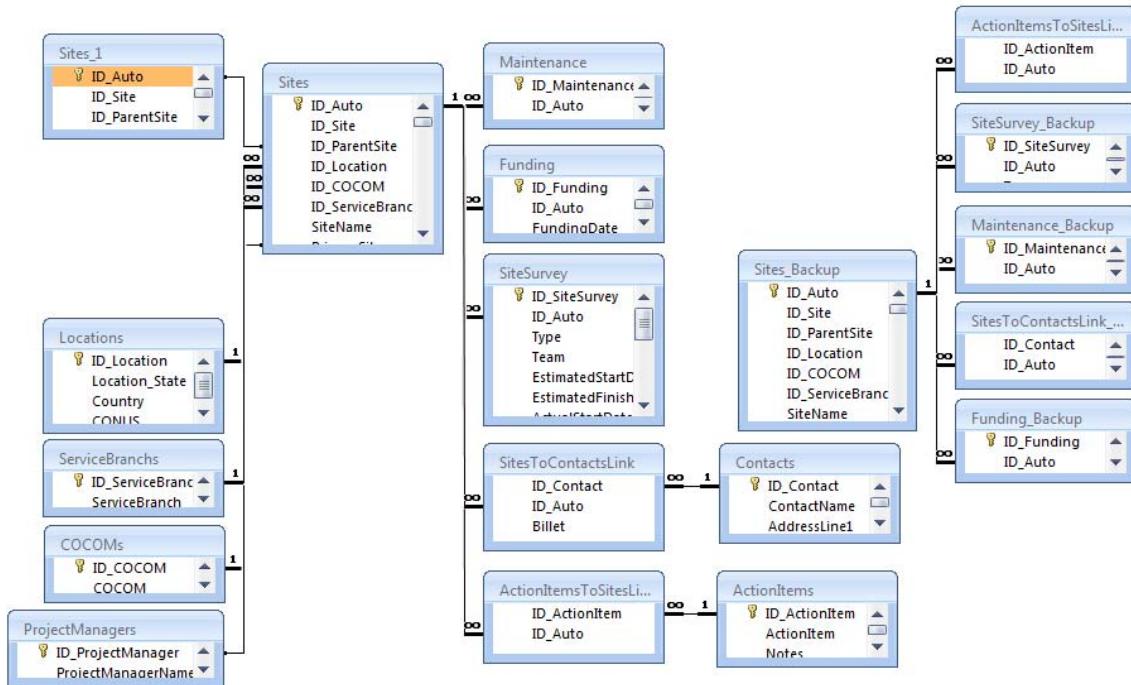


Figure 22. Access Key-Based Relationship Diagram

## **A. ActionItems**

Allows viewing of interoffice Action Items linked to various sites. The inclusion of this table was in preparation to allow for the linking to an outside Action Item database. Currently it only allows for the linkage to the Sites table and note taking.

### **User Permissions**

- Read only.

### **Primary Key**

- ID\_ActionItem

### **Cascading Update/Delete Relationships**

- ActionItemsToSitesLink

## **B. ActionItemsToSitesLink**

Allows for a many-to-many link between ActionItems and Sites.

### **User Permissions**

- Add/Update/Delete – However, within the GUI only deletions of a site will remove the linked records within this table.

### **Foreign Keys**

- ID\_ActionItem – ActionItems
- ID\_Auto – Sites

## **C. ActionItemsToSitesLink\_Backup**

Used to backup the linkage between ActionItems and Sites, if a site record is deleted.

### **User Permissions**

- Add/Update/Delete – However, within the GUI this is done indirectly through Visual Basic coded events.

### **Foreign Keys**

- ID\_ActionItem – ActionItems
- ID\_Auto – Sites\_Backup

## **D. COCOMs**

A listing of all of the COCOMs where there are sites established. Allows for filtering, sorting, and grouping of the site records in the Sites table.

### **User Permissions**

- Read Only.

**Primary Key**

- ID\_COCOM

**E. Contacts**

A listing of personnel allowing viewing of site contact information for each site.

**User Permissions**

- Read only.

**Primary Key**

- ID\_Contact

**F. Funding**

A listing of funding history information for a site.

**User Permissions**

- Add/Update/Delete.

**Primary Key**

- ID\_Funding

**Foreign Keys**

- ID\_Auto - Sites

**G. Funding\_Backup**

Used to backup the funding history for a deleted site.

**User Permissions**

- Add/Update/Delete – However, within the GUI this is done indirectly through Visual Basic coded events.

**Primary Key**

- ID\_Funding

**Foreign Keys**

- ID\_Auto – Sites\_Backup

**H. Locations**

A listing of all of the Locations where there are sites established. Allows for filtering, sorting, and grouping of the site records in the Sites table.

**User Permissions**

- Read Only.

**Primary Key**

- ID\_Location

**I. Maintenance**

A listing of maintenance history information for a site.

**User Permissions**

- Add/Update/Delete.

**Primary Key**

- ID\_Maint

**Foreign Keys**

- ID\_Auto - Sites

**J. Maintenance\_Backup**

Used to backup the maintenance history for a deleted site.

**User Permissions**

- Add/Update/Delete – However, within the GUI this is done indirectly through Visual Basic coded events.

**Primary Key**

- ID\_Maint

**Foreign Keys**

- ID\_Auto – Sites\_Backup

**K. ProjectManagers**

A listing of all of the Program Managers. Allows for filtering, sorting, and grouping of the site records in the Sites table.

**User Permissions**

- Read Only.

**Primary Key**

- ID\_ProjectManger

**L. ServiceBranchs**

A listing of all of the Service Branches that have sponsored sites. Allows for filtering, sorting, and grouping of the site records in the Sites table.

**User Permissions**

- Read Only.

### **Primary Key**

- ID\_ServiceBranch

## **M. Sites**

The Sites Table is the core of the DBIDS database, maintaining the information for each individual site. Special note on ID\_Site field is used to delineate between an actual site and a pending site. A “None” value will filter a site to pending, and a number value will identify it as a current site.

### **User Permissions**

- Add/Update/Delete.

### **Primary Key**

- ID\_Auto

### **Foreign Keys**

- ID\_ParentSite – A link back to sites to link children and parents, currently only used in the forms view.
- ID\_Location – Locations
- ID\_COCOM - COCOMs
- ID\_ServiceBranch – ServiceBranchs

### **Cascading Update/Delete Relationships**

There are five linked tables that when a delete is performed in Sites it will cause the deletion of linked records within them.

- Funding
- Maintenance
- SiteSurvey
- SitesToContactsLink
- ActionItemsToSitesLink

## **N. Sites\_Backup**

Used to backup a deleted site.

### **User Permissions**

- Add/Update/Delete – However, within the GUI this is done indirectly through Visual Basic coded events. Only users with Full Control User Permissions can navigate to restore or permanently remove a site.

### **Primary Key**

- ID\_Auto

### **Foreign Keys**

- ID\_ParentSite – A link back to sites to link children and parents, currently only used in the forms view.
- ID\_Location – Locations
- ID\_COCOM - COCOMs
- ID\_ServiceBranch – ServiceBranches

### **Cascading Update/Delete Relationships**

There are five linked tables that when a delete is performed in Sites it will cause the deletion of linked records within them.

- Funding
- Maintenance
- SiteSurvey
- SitesToContactsLink
- ActionItemsToSitesLink

## **O. SitesToContactsLink**

Allows for a many-to-many link between Contacts and Sites. Additionally identifies the billet of the contact in relation to the site.

### **User Permissions**

- Add/Update/Delete – However, within the GUI only deletions of a site will remove the linked records within this table.

### **Foreign Keys**

- ID\_Contact – Contacts
- ID\_Auto – Sites

## **P. SitesToContactsLink\_Backup**

Used to backup the linkage between Contacts and Sites, if a site record is deleted.

### **User Permissions**

- Add/Update/Delete – However, within the GUI this is done indirectly through Visual Basic coded events.

### **Foreign Keys**

- ID\_Contact – Contacts
- ID\_Auto – Sites\_Backup

## **Q. SiteSurvey**

A listing of site survey history information for a site.

#### **User Permissions**

- Add/Update/Delete.

#### **Primary Key**

- ID\_SiteSurvey

#### **Foreign Keys**

- ID\_Auto - Sites

### **R. SiteSurvey\_Backup**

Used to backup the maintenance history for a deleted site.

#### **User Permissions**

- Add/Update/Delete – However, within the GUI this is done indirectly through Visual Basic coded events.

#### **Primary Key**

- ID\_SiteSurvey

#### **Foreign Keys**

- ID\_Auto – Sites\_Backup

### **S. tblOpenReports**

The tblOpenReports table is used to dynamically build the reports filter form within the GUI. It allows for the selection of a report menu title, the targeted report name, allowing for additional filter fields and additional filter clauses. Allows the use of a single report form that is constructed dynamically whenever that form is opened.

#### **User Permissions**

- Read Only.

#### **Primary Key**

- ReportID

### **T. tbIPMFilters**

The tbIPMFilters table is used to dynamically build the report and form filters based on the logged in user. It allows for the creation of filters based on COCOM and or Service Branch. All users that have access to the database must have an entry in this table.

#### **User Permissions**

- Read Only.

### **Primary Key**

- User

## **U.   tblSwitchboardItems**

The `tblSwitchboardItems` table is used to dynamically build the menus within the GUI. It allows for the selection of a menu title, menu item label, one of thirteen different commands to be executed on selection, associated arguments for the command and whether to show the menu item based on the user's permission group. Allows the use of a single report form that is constructed dynamically whenever that form is opened.

The possible commands are:

- Switchboard – Used with the internal VB Code to identify a menu set.
- Goto Switchboard
- Open Form (Add)
- Open Form (Browse)
- Open Form (Datasheet)
- Open Report
- Open Report (Switchboard)
- Toggle Shift Bypass
- Exit Application
- Customize Switchboard – Currently not used within this GUI
- Run Macro – Currently not used within this GUI
- Run Code – Currently not used within this GUI
- Open Page – Currently not used within this GUI

### **User Permissions**

- Read Only.

### **Primary Key**

- SwitchboardID
- ItemNumber

## **V.   tblUnitInfo**

The `tblUnitInfo` is used internally by the VB Code, to allow for the removal of all users from the database.

### **User Permissions**

- Read Only.

## VI. QUERIES

The queries in the database are listed here. For each query, the following information is listed:

- SQL
- Dependencies

To view all table relationships refer to Figure 22

### A. qryAuditTrailClear

An update query to clear all the audit trail entries from the Sites tables.

#### SQL

```
UPDATE Sites SET Sites.AuditTrail = ""  
WHERE ((Sites.AuditTrail) <> "");
```

#### Dependencies

- Tables
  - Sites

### B. qryAuditTrailView

Used to find all sites with audit trail entries, whether the entries are in Sites, Funding, Maintenance, or SiteSurvey.

#### SQL

```
SELECT DISTINCTROW Sites.ID_Auto, Sites.ID_Site, Sites.SiteName,  
qryLocationList.Location, Sites.CommandName, COCOMs.COCOM, Sites.ID_COCOM,  
Sites.AuditTrail  
FROM (((COCOMs INNER JOIN (qryLocationList INNER JOIN Sites ON  
qryLocationList.ID_Location = Sites.ID_Location) ON COCOMs.ID_COCOM = Sites.ID_COCOM)  
LEFT JOIN Funding ON Sites.ID_Auto = Funding.ID_Auto) LEFT JOIN Maintenance ON  
Sites.ID_Auto = Maintenance.ID_Auto) LEFT JOIN SiteSurvey ON Sites.ID_Auto =  
SiteSurvey.ID_Auto  
WHERE (((Sites.AuditTrail)<>"")) OR (((SiteSurvey.AuditTrail)<>"")) OR  
(((Funding.AuditTrail)<>"")) OR (((Maintenance.AuditTrail)<>""));
```

#### Dependencies

- Tables
  - Funding
  - Maintenance
  - Sites
  - SiteSurvey
  - COCOMs
- Queries
  - qryLocationList

### C. qryBackupDeletedActionItemLinks

An insert query to copy the links, between Sites and ActionItems, to a backup table when a site is deleted.

#### SQL

```
INSERT INTO ActionItemsToSitesLink_Backup ( ID_Auto, ID_ActionItem )
SELECT ActionItemsToSitesLink.ID_Auto, ActionItemsToSitesLink.ID_ActionItem
FROM ActionItemsToSitesLink
WHERE (((ActionItemsToSitesLink.ID_Auto) = Forms!frmTargetID.TargetID.Value));
```

#### Dependencies

- Tables
  - ActionItemsToSitesLink
  - ActionItemsToSitesLink\_Backup
- Forms
  - frmTargetID

### D. qryBackupDeletedContactLinks

An insert query to copy the links, between Sites and Contacts, to a backup table when a site is deleted.

#### SQL

```
INSERT INTO SitesToContactsLink_Backup ( ID_Auto, ID_Contact, Billet )
SELECT SitesToContactsLink.ID_Auto, SitesToContactsLink.ID_Contact,
SitesToContactsLink.Billet
FROM SitesToContactsLink
WHERE (((SitesToContactsLink_Backup.ID_Auto) = Forms!frmTargetID.TargetID.Value));
```

#### Dependencies

- Tables
  - SitesToContactsLink
  - SitesToContactsLink\_Backup
- Forms
  - frmTargetID

### E. qryBackupDeletedFunding

An insert query to copy the funding history for a site to a backup table when a site is deleted.

#### SQL

```
INSERT INTO Funding_Backup ( ID_Auto, FundingDate, Reason, Required, Received, Source,
Notes, AuditTrail )
SELECT Funding.ID_Auto, Funding.FundingDate, Funding.Reason, Funding.Required,
Funding.Received, Funding.Source, Funding.Notes, Funding.AuditTrail
FROM Funding
WHERE (((Funding.ID_Auto) = Forms!frmTargetID.TargetID.Value));
```

#### Dependencies

- Tables
  - Funding
  - Funding\_Backup

- Sites
- SiteSurvey
- Forms
  - frmTargetID

## F. qryBackupDeletedMaint

An insert query to copy the maintenance history for a site to a backup table when a site is deleted.

### SQL

```
INSERT INTO Maintenance_Backup ( ID_Auto, MaintenanceDate, Type, Notes, AuditTrail )
SELECT Maintenance.ID_Auto, Maintenance.MaintenanceDate, Maintenance.Type,
Maintenance.Notes, Maintenance.AuditTrail
FROM Maintenance
WHERE ((Maintenance.ID_Auto) = Forms!frmTargetID.TargetID.Value);
```

### Dependencies

- Tables
  - Maintenance
  - Maintenance\_Backup
- Forms
  - frmTargetID

## G. qryBackupDeletedSite

An insert query to copy site information to a backup table when a site is deleted.

### SQL

```
INSERT INTO Sites_Backup ( ID_Auto, ID_Site, SiteName, PrimarySite, ID_ParentSite,
City, ID_Location, ID_COCOM, ID_ServiceBranch, CommandName, SvcMOUReceivedDate,
SSMChecklistDate, VPNChecklistDate, SiteIDAssigned, SiteSSAALead, SiteSSAADate,
EquipOrderDate, EquipReceivedTELOSDate, EquipBuildTELOSStartDate,
EquipBuildTELOSDuration, EquipReceivedSiteDate, DIPMSTELOSEntryDate,
RegStationInstallDate, RegStationInstallDuration, GateStationInstallDate,
GateStationInstallDuration, FieldingTeam, VPNConnectDate, DEERSPreLoadDate,
ConfigTrainingDate, ConfigTrainingDuration, ConfigTrainingTeam, SiteIOCDate,
CurrentCodeBase, CurrentSystemVersionNumber, CurrentHHSVersionNumber,
CurrentHandHeldVersionNumber, CurrentServerLocationSite, NextCodeBase,
NextSystemVersionNumber, NextHHSVersionNumber, NextHandHeldVersionNumber,
NextServerLocationSite, DeployDate, RegistrationPlanned, RegistrationInstalled,
LEOPlanned, LEOInstalled, GatesPlanned, GatesInstalled, MobileKitsPlanned,
MobileKitsInstalled, VisitorCenterPlanned, VisitorCenterInstalled, ServerOS,
DatabaseVersion, MaintYN, [Number], Vendors, TotalArchivedCurrent, TotalCurrent,
DBIDSCards, ProjMgr, Notes, ProjectID, ManPowerEffecientGateInstalled,
ManPowerEffecientGatePlanned, OracleLicenseVersion, MaintStartDate,
ExtraHandheldsPlanned, ExtraHandheldsInstalled, AuditTrail )
SELECT Sites.ID_Auto, Sites.ID_Site, Sites.SiteName, Sites.PrimarySite,
Sites.ID_ParentSite, Sites.City, Sites.ID_Location, Sites.ID_COCOM,
Sites.ID_ServiceBranch, Sites.CommandName, Sites.SvcMOUReceivedDate,
Sites.SSMChecklistDate, Sites.VPNChecklistDate, Sites.SiteIDAssigned,
Sites.SiteSSAALead, Sites.SiteSSAADate, Sites.EquipOrderDate,
Sites.EquipReceivedTELOSDate, Sites.EquipBuildTELOSStartDate,
Sites.EquipBuildTELOSDuration, Sites.EquipReceivedSiteDate, Sites.DIPMSTELOSEntryDate,
Sites.RegStationInstallDate, Sites.RegStationInstallDuration,
Sites.GateStationInstallDate, Sites.GateStationInstallDuration, Sites.FieldingTeam,
Sites.VPNConnectDate, Sites.DEERSPreLoadDate, Sites.ConfigTrainingDate,
Sites.ConfigTrainingDuration, Sites.ConfigTrainingTeam, Sites.SiteIOCDate,
Sites.CurrentCodeBase, Sites.CurrentSystemVersionNumber,
```

```

Sites.CurrentHHServerVersionNumber, Sites.CurrentHandHeldVersionNumber,
Sites.CurrentServerLocationSite, Sites.NextCodeBase, Sites.NextSystemVersionNumber,
Sites.NextHHServerVersionNumber, Sites.NextHandHeldVersionNumber,
Sites.NextServerLocationSite, Sites.DeployDate, Sites.RegistrationPlanned,
Sites.RegistrationInstalled, Sites.LEOPlanned, Sites.LEOInstalled, Sites.GatesPlanned,
Sites.GatesInstalled, Sites.MobileKitsPlanned, Sites.MobileKitsInstalled,
Sites.VisitorCenterPlanned, Sites.VisitorCenterInstalled, Sites.ServerOS,
Sites.DatabaseVersion, Sites.MaintYN, Sites.Number, Sites.Vendors,
Sites.TotalArchivedCurrent, Sites.TotalCurrent, Sites.DBIDSCards, Sites.ProjMgr,
Sites.Notes, Sites.ProjectID, Sites.ManPowerEffecientGateInstalled,
Sites.ManPowerEffecientGatePlanned, Sites.OracleLicenseVersion, Sites.MaintStartDate,
Sites.ExtraHandheldsPlanned, Sites.ExtraHandheldsInstalled,
Forms!frmTargetID!TargetAuditTrail AS AuditTrail
FROM Sites
WHERE (((Sites.ID_Auto) = Forms!frmTargetID.TargetID.Value));

```

### Dependencies

- Tables
  - Sites
  - Sites\_Backup
- Forms
  - frmTargetID

## H. qryBackupDeletedSiteSurvey

An insert query to copy the site survey history for a site, to a backup table when a site is deleted.

### SQL

```

INSERT INTO SiteSurvey_Backup ( ID_Auto, Type, Team, EstimatedStartDate,
EstimatedFinishDate, ActualStartDate, ActualFinishDate, CurrentStatus, InitialReview,
ReturnedTelos, RevisedReport, ROMReceived, DeliveredCustomer, Notes, Report,
AuditTrail )
SELECT SiteSurvey.ID_Auto, SiteSurvey.Type, SiteSurvey.Team,
SiteSurvey.EstimatedStartDate, SiteSurvey.EstimatedFinishDate,
SiteSurvey.ActualStartDate, SiteSurvey.ActualFinishDate, SiteSurvey.CurrentStatus,
SiteSurvey.InitialReview, SiteSurvey.ReturnedTelos, SiteSurvey.RevisedReport,
SiteSurvey.ROMReceived, SiteSurvey.DeliveredCustomer, SiteSurvey.Notes,
SiteSurvey.Report, SiteSurvey.AuditTrail
FROM SiteSurvey
WHERE (((SiteSurvey.ID_Auto) = Forms!frmTargetID.TargetID.Value));

```

### Dependencies

- Tables
  - SiteSurvey
  - SiteSurvey\_Backup
- Forms
  - frmTargetID

## I. qryBranchListFiltered

Used to allow for dynamic filtering within the VB Code for the service branch list on the Report Switchboard based on the user filters.

### SQL

```

SELECT DISTINCT ServiceBranches.ID_ServiceBranch, ServiceBranches.ServiceBranch,
qrySiteRelationships.ID_COCOM
FROM qrySiteRelationships INNER JOIN ServiceBranches ON
qrySiteRelationships.ID_ServiceBranch = ServiceBranches.ID_ServiceBranch;

```

### Dependencies

- Tables
  - ServiceBranchs
- Queries
  - qrySiteRelationships

## J. qryCommandList

Used to allow for dynamic filtering within the VB Code for the Command lists on the Report Switchboard based on the user filters.

### SQL

```
SELECT DISTINCT Sites.CommandName AS Command, COCOMs.COCOM,
ServiceBranchs.ServiceBranch AS Service
FROM ServiceBranchs INNER JOIN (COCOMs INNER JOIN Sites ON COCOMs.ID_COCOM =
Sites.ID_COCOM) ON ServiceBranchs.ID_ServiceBranch = Sites.ID_ServiceBranch
ORDER BY Sites.CommandName;
```

### Dependencies

- Tables
  - COCOMs
  - ServiceBranchs
  - Sites

## K. qryContactsList

Used for a drop down list of Contacts.

### SQL

```
SELECT Contacts.ID_Contact, Contacts.ContactName
FROM Contacts
ORDER BY Contacts.ContactName;
```

### Dependencies

- Tables
  - Contacts

## L. qryContactsReportByContact

Used by a report to list contact and site information grouped by Contacts.

### SQL

```
SELECT COCOMs.ID_COCOM, COCOMs.COCOM, ServiceBranchs.ID_ServiceBranch,
ServiceBranchs.ServiceBranch, qryLocationList.ID_Location, qryLocationList.Location,
Sites.ID_Site, qrySiteID_Name_Location.Site, SitesToContactsLink.Billet,
Contacts.ID_Contact, Contacts.ContactName, Contacts.Email, Contacts.CommPhone1,
Contacts.CommPhone2, Contacts.DSN1, Contacts.DSN2
FROM (ServiceBranchs RIGHT JOIN (COCOMs RIGHT JOIN (qryLocationList RIGHT JOIN (Sites
LEFT JOIN qrySiteID_Name_Location ON Sites.ID_Auto = qrySiteID_Name_Location.ID_Auto)
ON qryLocationList.ID_Location = Sites.ID_Location) ON COCOMs.ID_COCOM =
Sites.ID_COCOM) ON ServiceBranchs.ID_ServiceBranch = Sites.ID_ServiceBranch) RIGHT
JOIN (Contacts LEFT JOIN SitesToContactsLink ON Contacts.ID_Contact =
SitesToContactsLink.ID_Contact) ON Sites.ID_Auto = SitesToContactsLink.ID_Auto;
```

### Dependencies

- Tables

- COCOMs
- Contacts
- ServiceBranchs
- Sites
- SitesToContactsLink
- Queries
  - qryLocationList
  - qrySiteID\_Name\_Location

## M. qryContactsReportBySite

Used by a report to list contact and site information grouped by Sites.

### SQL

```
SELECT COCOMs.ID_COCOM, COCOMs.COCOM, ServiceBranchs.ID_ServiceBranch,
ServiceBranchs.ServiceBranch, qryLocationList.ID_Location, qryLocationList.Location,
SitesToContactsLink.ID_Auto, Sites.ID_Site, Sites.SiteName,
SitesToContactsLink.Billet, Contacts.ContactName, Contacts.Email, Contacts.CommPhone1,
Contacts.CommPhone2, Contacts.DSN1, Contacts.DSN2, Contacts.ShipToLine1,
Contacts.ShipToLine2, Contacts.ShipToLine3, Contacts.ShipToCity, Contacts.ShipToState,
Contacts.ShipToZip, Contacts.AddressLine1, Contacts.AddressLine2,
Contacts.AddressLine3, Contacts.AddressCity, Contacts.AddressState,
Contacts.AddressZip
FROM (ServiceBranchs RIGHT JOIN (COCOMs RIGHT JOIN (qryLocationList INNER JOIN Sites
ON qryLocationList.ID_Location = Sites.ID_Location) ON COCOMs.ID_COCOM =
Sites.ID_COCOM) ON ServiceBranchs.ID_ServiceBranch = Sites.ID_ServiceBranch) LEFT JOIN
(Contacts RIGHT JOIN SitesToContactsLink ON Contacts.ID_Contact =
SitesToContactsLink.ID_Contact) ON Sites.ID_Auto = SitesToContactsLink.ID_Auto
WHERE ((Sites.ID_Site)<>"None");
```

### Dependencies

- Tables
  - COCOMs
  - Contacts
  - ServiceBranchs
  - Sites
  - SitesToContactsLink
- Queries
  - qryLocationList
  - qrySiteID\_Name\_Location

## N. qryContactToSitesList

Used to list all site relationships for a contact.

### SQL

```
SELECT SitesToContactsLink.ID_Contact, SitesToContactsLink.ID_Auto,
SitesToContactsLink.Billet
FROM Sites RIGHT JOIN SitesToContactsLink ON Sites.ID_Auto =
SitesToContactsLink.ID_Auto;
```

### Dependencies

- Tables
  - Sites
  - SitesToContactsLink

## O. qryDeleteRestoredBackup

Used to remove a site from the site backup table.

### SQL

```
DELETE Sites_Backup.ID_Auto
FROM Sites_Backup
WHERE (((Sites_Backup.ID_Auto) = [Forms]![SiteBackup]![ID_Auto]));
```

### Dependencies

- Tables
  - Sites\_Backup
- Forms
  - SitesBackup

## P. qryFundingAudit

Used to list all funding history containing an audit trail.

### SQL

```
SELECT Funding.ID_Auto, Funding.FundingDate, Funding.AuditTrail
FROM Funding
WHERE ((Funding.AuditTrail) <> ""));
```

### Dependencies

- Tables
  - Funding

## Q. qryFundingAuditTrailClear

Used to clear all funding history audit trail entries.

### SQL

```
UPDATE Funding SET Funding.AuditTrail = ""
WHERE ((Funding.AuditTrail) <> ""));
```

### Dependencies

- Tables
  - Funding

## R. qryLocationList

Used to list combine location information for several drop down list and queries.

### SQL

```
SELECT Locations.ID_Location, [Location_State] & IIf([Location_State] =
[Country],",", " & [Country]) AS Location, Locations.Country,
Locations.Location_State
FROM Locations
ORDER BY Locations.Country, Locations.Location_State;
```

### Dependencies

- Tables
  - Locations

## S. qryLocationListFiltered

Used to allow for dynamic filtering within the VB Code for the location list on the Report Switchboard based on the user filters.

### SQL

```
SELECT DISTINCT qryLocationList.ID_Location, qryLocationList.Location,
qrySiteRelationships.ID_COCOM, qrySiteRelationships.ID_ServiceBranch,
qryLocationList.Country, qryLocationList.Location_State
FROM qryLocationList INNER JOIN qrySiteRelationships ON qryLocationList.ID_Location =
qrySiteRelationships.ID_Location
ORDER BY qryLocationList.Country, qryLocationList.Location_State;
```

### Dependencies

- Queries
  - qryLocationList
  - qrySiteRelationships

## T. qryMaintAuditTrailClear

Used to clear all maintenance history audit trail entries.

### SQL

```
UPDATE Maintenance SET Maintenance.AuditTrail = ""
WHERE ((Maintenance.AuditTrail) <> "");
```

### Dependencies

- Tables
  - Maintenance

## U. qryMaintenanceAudit

Used to clear all maintenance history audit trail entries.

### SQL

```
SELECT Maintenance.ID_Auto, Maintenance.MaintenanceDate, Maintenance.AuditTrail
FROM Maintenance
WHERE ((Maintenance.AuditTrail) <> "");
```

### Dependencies

- Tables
  - Maintenance

## V. qryMaintReport

Used to list all official sites and their current maintenance status.

### SQL

```
SELECT COCOMs.ID_COCOM, COCOMs.COCOM, Locations.ID_Location, Locations.Location_State,
ServiceBranchs.ID_ServiceBranch, ServiceBranchs.ServiceBranch, Sites.ID_Site,
Sites.SiteName, IIf(Sites!MaintYN, "Yes", "No") AS Maint, Sites.MaintStartDate
FROM ServiceBranchs INNER JOIN (Locations INNER JOIN (COCOMs INNER JOIN Sites ON
COCOMs.ID_COCOM = Sites.ID_COCOM) ON Locations.ID_Location = Sites.ID_Location) ON
ServiceBranchs.ID_ServiceBranch = Sites.ID_ServiceBranch
WHERE (((Sites.ID_Site) <> "None"));
```

## Dependencies

- Tables
  - COCOMs
  - Locations
  - ServiceBranchs
  - Sites

## W. qryParentSites

Used to list all primary sites in a drop down list.

### SQL

```
SELECT Sites.ID_Auto, [SiteName] & " - " & [Location] AS Site
FROM Sites INNER JOIN qryLocationList ON Sites.ID_Location =
qryLocationList.ID_Location
WHERE (((Sites.PrimarySite) = True))
ORDER BY [SiteName] & " - " & [Location];
```

## Dependencies

- Tables
  - Sites
- Queries
  - qryLocationList

## X. qryPendingSiteFormFilter

Used to allow for dynamic filtering within the VB Code for the pending sites based on the user filters.

### SQL

```
SELECT Sites.*
FROM Sites
WHERE (((Sites.ID_Site) = "None"))
ORDER BY Sites.SiteName;
```

## Dependencies

- Tables
  - Sites

## Y. qryPendingSiteReport

Used to allow for dynamic filtering within the VB Code for pending sites on the Report Switchboard based on the user filters.

### SQL

```
SELECT COCOMs.ID_COCOM, qryLocationList.ID_Location, qryLocationList.Location_State,
ServiceBranchs.ID_ServiceBranch, [SiteName] & " - " & [Location] AS Site,
Sites.ID_Auto
FROM ServiceBranchs INNER JOIN (COCOMs INNER JOIN (qryLocationList INNER JOIN Sites ON
qryLocationList.ID_Location = Sites.ID_Location) ON COCOMs.ID_COCOM = Sites.ID_COCOM)
ON ServiceBranchs.ID_ServiceBranch = Sites.ID_ServiceBranch
WHERE (((Sites.ID_Site) = "None"))
ORDER BY COCOMs.ID_COCOM;
```

## Dependencies

- Tables
  - COCOMs
  - Locations
  - ServiceBranchs
  - Sites
- Queries
  - qryLocationList

## Z. qryPlannedInstalledErrors

Used to identify sites with installed amounts larger than the planned amounts for the items on the Quick Stat report.

### SQL

```
SELECT Sites.ID_Auto, Sites.ID_Site, Sites.SiteName, ServiceBranchs.ServiceBranch, COCOMs.COCOM, Sites.RegistrationPlanned, Sites.RegistrationInstalled, Sites.LEOPlanned, Sites.LEOInstalled, Sites.GatesPlanned, Sites.GatesInstalled, Sites.MobileKitsPlanned, Sites.MobileKitsInstalled, Sites.VisitorCenterPlanned, Sites.VisitorCenterInstalled, Sites.ManPowerEffecientGatePlanned, Sites.ManPowerEffecientGateInstalled, Sites.ExtraHandheldsPlanned, Sites.ExtraHandheldsInstalled
FROM ServiceBranchs INNER JOIN (COCOMs INNER JOIN Sites ON COCOMs.ID_COCOM = Sites.ID_COCOM) ON ServiceBranchs.ID_ServiceBranch = Sites.ID_ServiceBranch
WHERE (((Sites.RegistrationInstalled) > [RegistrationPlanned])) OR
(((Sites.LEOInstalled) > [LEOPlanned])) OR (((Sites.GatesInstalled) > [GatesPlanned]))
OR (((Sites.MobileKitsInstalled) > [MobileKitsPlanned])) OR
(((Sites.VisitorCenterInstalled) > [VisitorCenterPlanned])) OR
(((Sites.ManPowerEffecientGateInstalled) > [ManPowerEffecientGatePlanned])) OR
(((Sites.ExtraHandheldsInstalled) > [ExtraHandheldsPlanned]))
ORDER BY Sites.SiteName;
```

## Dependencies

- Tables
  - COCOMs
  - Locations
  - ServiceBranchs
  - Sites

## AA. qryPMFilter

Used to build dynamic filters for several reports and forms based on the entries in the PM Filter table and the associated user currently logged into the database.

### SQL

```
SELECT tblPMFilters.User, tblPMFilters.COCOMFiltered,
tblPMFilters.ServiceBranchFiltered, IIf([COCOMFiltered] AND [ServiceBranchFiltered],"
(" & [COCOMFilter] & ") AND (" & [ServiceBranchFilter] & ")", IIf([COCOMFiltered],
[COCOMFilter], "")) AS PMFilter, IIf([COCOMFiltered], [COCOMFilter], "") AS COCOMOnly
FROM tblPMFilters;
```

## Dependencies

- Tables
  - tblPMFilters

## BB. qryQuickStatsReport

Used to prepare site data for consolidation in to the various forms of the Quick Stats Report.

### SQL

```
SELECT COCOMs.ID_COCOM, COCOMs.COCOM, Locations.ID_Location, Locations.Location_State, ServiceBranchs.ID_ServiceBranch, ServiceBranchs.ServiceBranch, Sites.SiteName, [PrimaryPlanned] + [SecondaryPlanned] AS TotalPlanned, [PrimaryInstalled] + [SecondaryInstalled] AS TotalInstalled, IIf([PrimarySite],1,0) AS PrimaryPlanned, IIf([PrimarySite], IIf([OperStatus] > 0, 1, 0), 0) AS PrimaryInstalled, IIf([PrimarySite], 0, 1) AS SecondaryPlanned, IIf([PrimarySite], 0, IIf([OperStatus] > 0, 1, 0)) AS SecondaryInstalled, Sites.PrimarySite, [RegistrationInstalled] + [LEOInstalled] + [GatesInstalled] + [MobileKitsInstalled] + [VisitorCenterInstalled] AS Installed, IIf([Planned] > 0, [Installed]/[Planned], 0) AS OperStatus, [RegistrationPlanned] + [LEOPlanned] + [GatesPlanned] + [MobileKitsPlanned] + [VisitorCenterPlanned] AS Planned, Sites.RegistrationPlanned, Sites.RegistrationInstalled, Sites.LEOPlanned, Sites.LEOInstalled, Sites.GatesPlanned, Sites.GatesInstalled, Sites.ManPowerEffecientGatePlanned, Sites.ManPowerEffecientGateInstalled, Sites.MobileKitsPlanned, Sites.MobileKitsInstalled, Sites.VisitorCenterPlanned, Sites.VisitorCenterInstalled, Sites.ExtraHandheldsPlanned, Sites.ExtraHandheldsInstalled, Sites.TotalArchivedCurrent, Sites.TotalCurrent, Sites.DBIDSCards
FROM ServiceBranchs INNER JOIN (Locations INNER JOIN (COCOMs INNER JOIN Sites ON COCOMs.ID_COCOM = Sites.ID_COCOM) ON Locations.ID_Location = Sites.ID_Location) ON ServiceBranchs.ID_ServiceBranch = Sites.ID_ServiceBranch;
```

### Dependencies

- Tables
  - COCOMs
  - Locations
  - ServiceBranchs
  - Sites

## CC. qryRestoreDeletedActionItemLinks

Used to restore the links between a Site and an Action Item, when restoring a deleted site from the backup tables.

### SQL

```
INSERT INTO ActionItemsToSitesLink ( ID_Auto, ID_ActionItem )
SELECT Forms!frmTargetID.TargetID.Value AS TargetID,
ActionItemsToSitesLink_Backup.ID_ActionItem
FROM ActionItemsToSitesLink_Backup
WHERE ((ActionItemsToSitesLink_Backup.ID_Auto) = Forms!SiteBackup.ID_Auto));
```

### Dependencies

- Tables
  - ActionItemsToSitesLink
  - ActionItemsToSitesLink\_Backup

## DD. qryRestoreDeletedContactLinks

Used to restore the links between a Site and a Contact, when restoring a deleted site from the backup tables.

## SQL

```
INSERT INTO SitesToContactsLink ( ID_Auto, ID_Contact, Billet )
SELECT Forms!frmTargetID.TargetID.Value AS TargetID,
SitesToContactsLink_Backup.ID_Contact, SitesToContactsLink_Backup.Billet
FROM SitesToContactsLink_Backup
WHERE (((SitesToContactsLink_Backup.ID_Auto) = Forms!SiteBackup.ID_Auto));
```

## Dependencies

- Tables
  - SitesToContactsLink
  - SitesToContactsLink\_Backup
- Forms
  - SiteBackup

## EE. qryRestoreDeletedFunding

Used to restore the Funding history for a Site when restoring that deleted site from the backup tables.

## SQL

```
INSERT INTO Funding ( ID_Auto, FundingDate, Reason, Required, Received, Source, Notes,
AuditTrail )
SELECT Forms!frmTargetID.TargetID.Value AS TargetID, Funding_Backup.FundingDate,
Funding_Backup.Reason, Funding_Backup.Required, Funding_Backup.Received,
Funding_Backup.Source, Funding_Backup.Notes, Funding_Backup.AuditTrail
FROM Funding_Backup
WHERE (((Funding_Backup.ID_Auto) = Forms!SiteBackup.ID_Auto));
```

## Dependencies

- Tables
  - Funding
  - Funding\_Backup
- Forms
  - SiteBackup

## FF. qryRestoreDeletedMaint

Used to restore the maintenance history for a site when restoring that deleted site from the backup tables.

## SQL

```
INSERT INTO Maintenance ( ID_Auto, MaintenanceDate, Type, Notes, AuditTrail )
SELECT Forms!frmTargetID.TargetID.Value AS TargetID,
Maintenance_Backup.MaintenanceDate, Maintenance_Backup.Type, Maintenance_Backup.Notes,
Maintenance_Backup.AuditTrail
FROM Maintenance_Backup
WHERE (((Maintenance_Backup.ID_Auto) = Forms!SiteBackup.ID_Auto));
```

## Dependencies

- Tables
  - Maintenance
  - Maintenance\_Backup
- Forms
  - SiteBackup

## GG. qryRestoreDeletedSite

Used to restore a deleted site from the backup tables.

### SQL

```
INSERT INTO Sites ( ID_Site, SiteName, PrimarySite, ID_ParentSite, City, ID_Location, ID_COCOM, ID_ServiceBranch, CommandName, SvcMOUReceivedDate, SSMChecklistDate, VPNChecklistDate, SiteIDAssigned, SiteSSAALead, SiteSSAADate, EquipOrderDate, EquipReceivedTELOSDate, EquipBuildTELOSStartDate, EquipBuildTELOSDuration, EquipReceivedSiteDate, DIPMSTELOSEntryDate, RegStationInstallDate, RegStationInstallDuration, GateStationInstallDate, GateStationInstallDuration, FieldingTeam, VPNCConnectDate, DEERSPreLoadDate, ConfigTrainingDate, ConfigTrainingDuration, ConfigTrainingTeam, SiteIOCDate, CurrentCodeBase, CurrentSystemVersionNumber, CurrentHHServerVersionNumber, CurrentHandHeldVersionNumber, CurrentServerLocationSite, NextCodeBase, NextSystemVersionNumber, NextHHServerVersionNumber, NextHandHeldVersionNumber, NextServerLocationSite, DeployDate, RegistrationPlanned, RegistrationInstalled, LEOPlanned, LEOInstalled, GatesPlanned, GatesInstalled, MobileKitsPlanned, MobileKitsInstalled, VisitorCenterPlanned, VisitorCenterInstalled, ServerOS, DatabaseVersion, MaintYN, [Number], Vendors, TotalArchivedCurrent, TotalCurrent, DBIDSCards, ProjMgr, Notes, ProjectID, ManPowerEffecientGateInstalled, ManPowerEffecientGatePlanned, OracleLicenseVersion, MaintStartDate, ExtraHandheldsPlanned, ExtraHandheldsInstalled, AuditTrail )  
SELECT Sites_Backup.ID_Site, Sites_Backup.SiteName, Sites_Backup.PrimarySite, Sites_Backup.ID_ParentSite, Sites_Backup.City, Sites_Backup.ID_Location, Sites_Backup.ID_COCOM, Sites_Backup.ID_ServiceBranch, Sites_Backup.CommandName, Sites_Backup.SvcMOUReceivedDate, Sites_Backup.SSMChecklistDate, Sites_Backup.VPNChecklistDate, Sites_Backup.SiteSSAALead, Sites_Backup.SiteSSAADate, Sites_Backup.EquipOrderDate, Sites_Backup.EquipReceivedTELOSDate, Sites_Backup.EquipBuildTELOSStartDate, Sites_Backup.EquipBuildTELOSDuration, Sites_Backup.EquipReceivedSiteDate, Sites_Backup.DIPMSTELOSEntryDate, Sites_Backup.RegStationInstallDate, Sites_Backup.RegStationInstallDuration, Sites_Backup.GateStationInstallDate, Sites_Backup.GateStationInstallDuration, Sites_Backup.FieldingTeam, Sites_Backup.VPNCConnectDate, Sites_Backup.DEERSPreLoadDate, Sites_Backup.ConfigTrainingDate, Sites_Backup.ConfigTrainingDuration, Sites_Backup.ConfigTrainingTeam, Sites_Backup.SiteIOCDate, Sites_Backup.CurrentCodeBase, Sites_Backup.CurrentSystemVersionNumber, Sites_Backup.CurrentHHServerVersionNumber, Sites_Backup.CurrentHandHeldVersionNumber, Sites_Backup.CurrentServerLocationSite, Sites_Backup.NextCodeBase, Sites_Backup.NextSystemVersionNumber, Sites_Backup.NextHHServerVersionNumber, Sites_Backup.NextHandHeldVersionNumber, Sites_Backup.NextServerLocationSite, Sites_Backup.DeployDate, Sites_Backup.RegistrationPlanned, Sites_Backup.RegistrationInstalled, Sites_Backup.LEOPlanned, Sites_Backup.MobileKitsPlanned, Sites_Backup.GatesPlanned, Sites_Backup.GatesInstalled, Sites_Backup.VisitorCenterPlanned, Sites_Backup.VisitorCenterInstalled, Sites_Backup.ServerOS, Sites_Backup.DatabaseVersion, Sites_Backup.MaintYN, Sites_Backup.Number, Sites_Backup.Vendors, Sites_Backup.TotalArchivedCurrent, Sites_Backup.TotalCurrent, Sites_Backup.DBIDSCards, Sites_Backup.ProjMgr, Sites_Backup.Notes, Sites_Backup.ProjectID, Sites_Backup.ManPowerEffecientGateInstalled, Sites_Backup.ManPowerEffecientGatePlanned, Sites_Backup.OracleLicenseVersion, Sites_Backup.MaintStartDate, Sites_Backup.ExtraHandheldsPlanned, Sites_Backup.ExtraHandheldsInstalled, "****RESTORING****" AS AuditTrail  
FROM Sites_Backup  
WHERE ((Sites_Backup.ID_Auto) = Forms!SiteBackup.ID_Auto));
```

### Dependencies

- Tables
  - Sites
  - Sites\_Backup
- Forms
  - SiteBackup

## HH. qryRestoreDeletedSiteSurvey

Used to restore the Site Survey History for a site when restoring that deleted site from the backup tables.

### SQL

```
INSERT INTO SiteSurvey ( ID_Auto, Type, Team, EstimatedStartDate, EstimatedFinishDate, ActualStartDate, ActualFinishDate, CurrentStatus, InitialReview, ReturnedTelos, RevisedReport, ROMReceived, DeliveredCustomer, Notes, Report, AuditTrail )
SELECT Forms!frmTargetID.TargetID AS TargetID, SiteSurvey_Backup.Type,
SiteSurvey_Backup.Team, SiteSurvey_Backup.EstimatedStartDate,
SiteSurvey_Backup.EstimatedFinishDate, SiteSurvey_Backup.ActualStartDate,
SiteSurvey_Backup.ActualFinishDate, SiteSurvey_Backup.CurrentStatus,
SiteSurvey_Backup.InitialReview, SiteSurvey_Backup.ReturnedTelos,
SiteSurvey_Backup.RevisedReport, SiteSurvey_Backup.ROMReceived,
SiteSurvey_Backup.DeliveredCustomer, SiteSurvey_Backup.Notes,
SiteSurvey_Backup.Report, SiteSurvey_Backup.AuditTrail
FROM SiteSurvey_Backup
WHERE ((SiteSurvey_Backup.ID_Auto) = Forms!SiteBackup.ID_Auto);
```

### Dependencies

- Tables
  - SiteSurvey
  - SiteSurvey\_Backup
- Forms
  - SiteBackup

## II. qryRestoringSiteAuditTrail

Used to restore the previous audit trail for a site and append a restored flag, when restoring a deleted site from the backup tables.

### SQL

```
UPDATE Sites SET Sites.AuditTrail = "***Restored*** " &
Forms!frmTargetID!TargetAuditTrail
WHERE ((Sites.ID_Auto) = Forms!frmTargetID!TargetID));
```

### Dependencies

- Tables
  - Sites
- Forms
  - frmTargetID

## JJ. qrySiteActionItems

Used to list Action Item information for a Site.

### SQL

```
SELECT ActionItemsToSitesLink.ID_Auto, ActionItems.ID_ActionItem,
ActionItems.ActionItem, ActionItems.Notes
FROM ActionItems INNER JOIN ActionItemsToSitesLink ON ActionItems.ID_ActionItem =
ActionItemsToSitesLink.ID_ActionItem
ORDER BY ActionItems.ActionItem DESC;
```

### Dependencies

- Tables

- ActionItems
- ActionItemsToSitesLink

## KK. qrySiteFormFilter

Used to allow for dynamic filtering within the VB Code for the sites based on the user filters.

### SQL

```
SELECT Sites.*
FROM Sites
WHERE ((Sites.ID_Site) <> "None")
ORDER BY Sites.SiteName;
```

### Dependencies

- Tables
- Sites

## LL. qrySiteID\_Name\_Location

Used to consolidate an edited nametag for sites, allowing for easier identification, by Site ID, Name, or Location.

### SQL

```
SELECT Sites.ID_Auto, Sites.ID_Site & " / " & [SiteName] & ":" & [Location_State] &
IF([Location_State] = [Country], "", " " & [Country]) AS Site, Sites.ID_COCOM,
Sites.ID_ServiceBranch, Sites.ID_Site
FROM Locations INNER JOIN Sites ON Locations.ID_Location = Sites.ID_Location
ORDER BY Sites.SiteName;
```

### Dependencies

- Tables
- Locations
- Sites

## MM. qrySiteRelationships

Used in the dynamic filtering of the Reports Switchboard.

### SQL

```
SELECT DISTINCT COCOMs.ID_COCOM, Locations.ID_Location,
ServiceBranchs.ID_ServiceBranch
FROM (COCOMs INNER JOIN (Locations INNER JOIN Sites ON Locations.ID_Location =
Sites.ID_Location) ON COCOMs.ID_COCOM = Sites.ID_COCOM) INNER JOIN ServiceBranchs ON
Sites.ID_ServiceBranch = ServiceBranchs.ID_ServiceBranch;
```

### Dependencies

- Tables
- COCOMs
- Locations
- Sites
- ServiceBranchs

## NN. qrySiteSurveyAudit

Used to list all Site Survey history containing an audit trail.

### SQL

```
SELECT SiteSurvey.ID_Auto, SiteSurvey.Type, SiteSurvey.ActualStartDate,
SiteSurvey.ActualFinishDate, SiteSurvey.AuditTrail
FROM SiteSurvey
WHERE (((SiteSurvey.AuditTrail) <> ""));
```

### Dependencies

- Tables

SiteSurvey

## OO. qrySiteSurveyAuditTrailClear

Used to clear all Site Survey history audit trail entries.

### SQL

```
UPDATE SiteSurvey SET SiteSurvey.AuditTrail = ""
WHERE (((SiteSurvey.AuditTrail) <> ""));
```

### Dependencies

- Tables
  - SiteSurvey

## PP. qrySiteSurveyReport

Used to list a selected group of Site Survey to audit the timeliness of the site survey process.

### SQL

```
SELECT COCOMs.COCOM, ServiceBranchs.ServiceBranch, Sites.ID_Site,
Locations.Location_State, Locations.Country, Locations.CONUS, Sites.SiteName,
SiteSurvey.Type, SiteSurvey.Team, SiteSurvey.EstimatedStartDate,
SiteSurvey.EstimatedFinishDate, DateDiff("d",[EstimatedStartDate],
[EstimatedFinishDate]) + 1 AS EstDuration, SiteSurvey.ActualStartDate,
SiteSurvey.ActualFinishDate, DateDiff("d", [ActualStartDate], [ActualFinishDate]) + 1
AS ActualDuration, SiteSurvey.CurrentStatus, SiteSurvey.InitialReview,
SiteSurvey.ReturnedTelos, SiteSurvey.RevisedReport, SiteSurvey.ROMReceived,
SiteSurvey.DeliveredCustomer, SiteSurvey.Notes, Locations.ID_Location,
COCOMs.ID_COCOM, ServiceBranchs.ID_ServiceBranch, DateAdd("d", 14, [ActualFinishDate])
AS TargetDate
FROM (ServiceBranchs INNER JOIN (Locations INNER JOIN (COCOMs INNER JOIN Sites ON
COCOMs.ID_COCOM = Sites.ID_COCOM) ON Locations.ID_Location = Sites.ID_Location) ON
ServiceBranchs.ID_ServiceBranch = Sites.ID_ServiceBranch) INNER JOIN SiteSurvey ON
Sites.ID_Auto = SiteSurvey.ID_Auto
WHERE (((SiteSurvey.Report) = Yes));
```

### Dependencies

- Tables
  - COCOMs
  - Locations
  - ServiceBranchs
  - Sites
  - SiteSurvey

## QQ. qrySiteToContactsList

Used to list all contacts for a site.

### SQL

```
SELECT SitesToContactsLink.ID_Contact, SitesToContactsLink.ID_Auto,
SitesToContactsLink.Billet
FROM Sites LEFT JOIN SitesToContactsLink ON Sites.ID_Auto =
SitesToContactsLink.ID_Auto;
```

### Dependencies

- Tables
  - Sites
  - SitesToContactsLink

## RR. qryViewPMFilters

Used to test a PMs filters.

### SQL

```
SELECT Sites.ID_Site, Sites.SiteName, COCOMs.ID_COCOM, COCOMs.COCOM,
ServiceBranchs.ID_ServiceBranch, ServiceBranchs.ServiceBranch
FROM ServiceBranchs INNER JOIN (Sites INNER JOIN COCOMs ON Sites.ID_COCOM =
COCOMs.ID_COCOM) ON ServiceBranchs.ID_ServiceBranch = Sites.ID_ServiceBranch
ORDER BY Sites.SiteName;
```

### Dependencies

- Tables
  - COCOMs
  - ServiceBranchs
  - Sites

THIS PAGE INTENTIONALLY LEFT BLANK

## VII. FORMS

The forms in the database are listed here. For each form, the following information is listed:

- Record Source – The query, SQL, table, or that the records on the form are pulled from.
- Dependencies – The query, SQL, or table that a list or drop-down list requires to pull information from, or the subforms within this form.
- Order By – The fields on the form that the records are sorted on.
- Events – All the VB Script Code associated to the form.
- Controls – Identifies if there are any linked tables that when a record is deleted from the main table it would remove any linked records from the linked tables.

### A. ActionItems

#### Record Source

- ActionItems

#### Dependencies

- Subforms
  - Sites: subformActionItemSites

#### Events

- Clear Button Click

```
Private Sub btnClear_Click()
    Me.RecordSource = "ActionItems"
    Me.Search_Field.Value = ""
    Me.Filter = ""
    Me.FilterOn = False
End Sub
```

- Search Action Item Button Click

```
Private Sub btnActionItem_Click()
    Me.Filter = "ActionItem Like '*' & Me.Search_Field & '*'"
    Me.FilterOn = True
    If Me.Recordset.RecordCount = 0 Then
        Me.Filter = ""
        Me.FilterOn = True
        MsgBox "There are no records that meet that criteria!!"
    End If
End Sub
```

## **Controls**

- Action Item: ActionItems
- Notes: Notes
- Search Action Item(Button): btnActionItem\_Click
- Clear(Button): btnClear\_Click
- Search Field(Unlabeled): Unbound

## **B. AuditTrailView**

### **Record Source**

- qryAuditTrailView

### **Order By**

- ID\_COCOM
- Location
- SiteName

### **Dependencies**

- Subforms
  - Funding: subFormFundingAudit
  - Maintenance: subFormMaintenanceAudit
  - Site Survey: subFormSiteSurveyAudit

### **Events**

- Clear Button Click

```

•
Private Sub btnClear_Click()
On Error GoTo Err_btnClear_Click
    Dim stDocName As String
    DoCmd.SetWarnings (False)
    stDocName = "qryAuditTrailClear"
    DoCmd.OpenQuery stDocName, acNormal, acEdit
    stDocName = "qryFundingAuditTrailClear"
    DoCmd.OpenQuery stDocName, acNormal, acEdit
    stDocName = "qryMaintAuditTrailClear"
    DoCmd.OpenQuery stDocName, acNormal, acEdit
    stDocName = "qrySiteSurveyAuditTrailClear"
    DoCmd.OpenQuery stDocName, acNormal, acEdit

    DoCmd.SetWarnings (True)
    Me.Requery
    Me.Refresh
    If Me.Recordset.EOF Then
        Me.btnAddRecord.Enabled = False
        Me.btnAdd.Enabled = False
        Me.btnAddPrint.Enabled = False
    Else
        Me.btnAddRecord.Enabled = True
        Me.btnAdd.Enabled = True
        Me.btnAddPrint.Enabled = True
    End If
Exit_btnClear_Click:
    Exit Sub
Err_btnClear_Click:
    MsgBox Err.Description
    Resume Exit_btnClear_Click
End Sub

```

#### • Print Button Click

```

Private Sub btnPrint_Click()
On Error GoTo Err_btnPrint_Click
    Dim stDocName As String
    stDocName = "rptAuditTrail"
    DoCmd.OpenReport stDocName, acViewPreview
Exit_btnPrint_Click:
    Exit Sub
Err_btnPrint_Click:
    MsgBox Err.Description
    Resume Exit_btnPrint_Click
End Sub

```

#### • Goto Record Button Click

```

Private Sub btnGotoRecord_Click()
On Error GoTo Err_btnGotoRecord_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    If Me![ID_Site] = "None" Then
        stDocName = "PendingSites"
    Else
        stDocName = "Sites"
    End If

    stLinkCriteria = "[ID_Auto] = " & Me![ID_Auto]
    DoCmd.OpenForm stDocName, , , stLinkCriteria, , , stLinkCriteria

Exit_btnGotoRecord_Click:
    Exit Sub
Err_btnGotoRecord_Click:
    MsgBox Err.Description
    Resume Exit_btnGotoRecord_Click
End Sub

```

- On Current

```
Private Sub Form_Current()
    If Me.Recordset.EOF Then
        Me.btnAddRecord.Enabled = False
        Me.btnClear.Enabled = False
        Me.btnPrint.Enabled = False
    Else
        Me.btnAddRecord.Enabled = True
        Me.btnClear.Enabled = True
        Me.btnPrint.Enabled = True
    End If
End Sub
```

- On Open

```
Private Sub Form_Open(Cancel As Integer)
    If Me.Recordset.EOF Then
        Me.btnAddRecord.Enabled = False
        Me.btnClear.Enabled = False
        Me.btnPrint.Enabled = False
    Else
        Me.btnAddRecord.Enabled = True
        Me.btnClear.Enabled = True
        Me.btnPrint.Enabled = True
    End If
End Sub
```

## Controls

- ID\_Auto(Hidden): ID\_Auto
- Audit Trail(Unlabeled): AuditTrail
- COCOM: COCOM
- Location: Location
- Site ID: ID\_Site
- Site Name: SiteName
- Command: Command Name
- Clear All Audit Trail Entries(Button): btnClear\_Click
- Goto Record(Button): btnGotoRecord\_Click
- Print(Button): btnPrint\_Click

## C. COCOMs

### Record Source

- COCOMs

### Controls

- COCOM: COCOM

## D. ContactInformationByContact

### Record Source

- Contacts

### Dependencies

- Subforms
  - Sites: subFormContactInformationToSite

### Events

- Clear Button Click

```
Private Sub btnClear_Click()
    Me.Search_Field.Value = ""
    Me.Filter = ""
    Me.FilterOn = False
End Sub

• Search Name Button Click

Private Sub btnSearchName_Click()
    Me.Filter = "[ContactName] Like '*" & Me.Search_Field & "*'"
    Me.FilterOn = True
    If Me.Recordset.RecordCount = 0 Then
        Me.Filter = ""
        Me.FilterOn = True
        MsgBox "There are no records that meet that criteria!!"
    End If
End Sub
```

- On Close

```
Private Sub Form_Close()
    If CurrentProject.AllForms("ContactInformationbySite").IsLoaded Then
        Forms![ContactInformationbySite].Requery
    End If
End Sub
```

## Controls

- Name: ContactName
- Email: Email
- Address Line 1: AddressLine1
- Address Line 2: AddressLine2
- Address Line 3: AddressLine3
- Address City: : AddressCity
- Address State: AddressState
- Address Zip: AddressZip
- Ship To Line 1: ShipToLine1
- Ship To Line 2: ShipToLine2
- Ship To Line 3: ShipToLine3
- Ship To City: ShipToCity
- Ship To State: ShipToState
- Ship To Zip: ShipToZip
- Comm Phone #1: CommPhone1
- Comm Phone # 2: CommPhone2
- DSN #1: DSN1
- DSN #2: DSN2
- Fax #: Fax
- Search Name(Button): btnSearchName\_Click
- Clear(Button): btnClear\_Click
- Search Field(Unlabeled): Unbound

## E. ContactInformationBySite

### Record Source

- Sites

### Dependencies

- Drop-Down Lists
  - Location: qryLocationList
- Subforms
  - Contacts: subFormSiteToContactInformation

### Order By

- SiteName

### Events

- On Clear Button Click

```
Private Sub btnClear_Click()
    Me.Search_Field.Value = ""
    Me.Filter = ""
    Me.FilterOn = False
End Sub
```

- On Search ID Button Click

```
Private Sub btnSearchID_Click()
    Me.Filter = "ID_Site Like '*' & Me.Search_Field & '*'"
    Me.FilterOn = True

    If Me.Recordset.RecordCount = 0 Then
        Me.Filter = ""
        Me.FilterOn = True
    End If
End Sub
```

```

        MsgBox "There are no records that meet that criteria!!"
    End If
End Sub

```

- On Search Name Click

```

Private Sub btnSearchName_Click()
    Me.Filter = "SiteName Like '*' & Me.Search_Field & '*'"
    Me.FilterOn = True

    If Me.Recordset.RecordCount = 0 Then
        Me.Filter = ""
        Me.FilterOn = True
        MsgBox "There are no records that meet that criteria!!"
    End If
End Sub

```

### Controls

- Site ID: ID\_Site
- Site: SiteName
- Command: Command Name
- Location: ID\_Location
- Search ID(Button): btnSearchID\_Click
- Search Name(Button): btnSearchName\_Click
- Clear(Button): btnClear\_Click
- Search Field(Unlabeled): Unbound

## F. frmExitMessage

### Events

- On Timer (1 Second Interval)

```

Private Sub Form_Timer()
    Me.Refresh
End Sub

```

### Controls

- Text Box(Unlabeled):="Time remaining " & Forms!frmHiddenManager!TimePassed & " Seconds."

## G. frmHiddenManager

### Record Source

- tblUnitInfo

### Events

- After Five Minute Warning Update

```

Private Sub FiveMinWarning_AfterUpdate()
    If Me.FiveMinWarning Then
        Me.TimePassed = 300
    End If
End Sub

```

- On Current

```

Private Sub Form_Current()
    Dim con As ADODB.Connection
    Dim rs As ADODB.Recordset
    Dim i, j As Integer
    i = 1
    j = 1

```

```

Set con = Application.CurrentProject.Connection
' The user roster is exposed as a provider-specific schema rowset
' in the Jet 4.0 OLE DB provider. You have to use a GUID to
' reference the schema, as provider-specific schemas are not
' listed in ADO's type library for schema rowsets
Set rs = con.OpenSchema(adSchemaProviderSpecific, , "{947bb102-5d43-11d1-bdbf-
c04fb92675}")
'Output the list of all users in the current database.
Me.Label1.Caption = rs.Fields(0).name
Me.Label2.Caption = rs.Fields(1).name
Me.Label3.Caption = rs.Fields(2).name
Me.Label4.Caption = rs.Fields(3).name
While (Not (rs.EOF)) And i < 11
    While j < 4
        Me("r" & i & "c" & (j + 1)).Visible = True
        If Not IsNull(rs.Fields(j).Value) Then
            Me("r" & i & "c" & (j + 1)).Caption = rs.Fields(j).Value
        End If
        j = j + 1
    Wend
    j = 0
    rs.MoveNext
    i = i + 1
Wend
' Close the recordset and the database.
rs.Close
Set rs = Nothing
Set con = Nothing
End Sub

```

- On Timer (5 Second Interval)

```

Private Sub Form_Timer()
On Error GoTo Err_ForceOut
Dim MsgBoxTxt, MsgBoxTxtTime As String

If Me.GetOut Then
    GoTo Err_ForceOut
ElseIf Me.FiveMinWarning Then
    If Me.TimePassed = 30 Then
        DoCmd.OpenForm "frmExitMessage", acNormal, , , acFormReadOnly
    ElseIf Me.TimePassed = 300 Then
        DoCmd.OpenForm "frmExitMessage", acNormal, , , acFormReadOnly
    End If
    Me.TimePassed = Me.TimePassed - 5
    If Me.TimePassed <= 0 Then GoTo Err_ForceOut
End If
Exit_ForceOut:
Me.Requery
Exit Sub
Err_ForceOut:
Me.GetOut = No
Me.FiveMinWarning = No
Me.TimePassed = 300
Application.Quit
End Sub

```

## Controls

- Edited Time Passed(Unlabeled): =[TimePassed] & " Seconds"
- Five Minute Warning: FiveMinWarning
- Force Out Now: GetOut
- Time Passed(Hidden): TimePassed

## H. frmOpenReports

### Record Source

- tblOpenReports

### Dependencies

- Drop-Down Lists
  - Branchs: ServiceBranchs
  - Locations: qryLocationList
  - COCOMs: COCOMs

### Events

- After All COCOMs Checkbox Update

```
Private Sub All_COCOMs_AfterUpdate()
'Make COCOM List unselectable when box for All is checked
If Me.All_COCOMs Then
    Dim varItem As Variant
    For Each varItem In Me.COCOM_List.ItemsSelected
        Me.COCOM_List.Selected(varItem) = False
    Next varItem
    Me.Branch_List.RowSource = "SELECT DISTINCT ID_ServiceBranch,
ServiceBranch FROM
qryBranchListFiltered WHERE " & MultiListItems(Me.COCOM_List, "[ID_COCOM]") &
GetCurrentUserFilter(False, False)
    Me.Location_List.RowSource = "SELECT DISTINCT ID_Location, Location, Country FROM
qryLocationListFiltered WHERE " & MultiListItems(Me.COCOM_List, "[ID_COCOM]") &
"AND " &
    MultiListItems(Me.Branch_List, "[ID_ServiceBranch]") & GetCurrentUserFilter(False,
False) & "
    ORDER BY Country, Location"
    Me.COCOM_List.Enabled = False

    Me.Branch_List.Enabled = False
    Me.All_Branchs.Value = True
    Me.Location_List.Enabled = False
    Me.All_Locations.Value = True

    Else
        Me.COCOM_List.Enabled = True
    End If
End Sub
```

- After All Locations Checkbox Update

```
Private Sub All_Locations_AfterUpdate()
'Make Location List unselectable when box for All is checked
If Me.All_Locations Then
    Dim varItem As Variant
    For Each varItem In Me.Location_List.ItemsSelected
        Me.Location_List.Selected(varItem) = False
    Next varItem
    Me.Location_List.Enabled = False
Else
    Me.Location_List.Enabled = True
End If
End Sub
```

- After All Branchs Checkbox Update

```
Private Sub All_Branchs_AfterUpdate()
'Make Branch List unselectable when box for All is checked
If Me.All_Branchs Then
    Dim varItem As Variant
    For Each varItem In Me.Branch_List.ItemsSelected
        Me.Branch_List.Selected(varItem) = False
    Next varItem
End Sub
```

```

        Next varItem
        Me.Location_List.RowSource = "SELECT DISTINCT ID_Location, Location, Country FROM
        qryLocationListFiltered WHERE " & MultiListItems(Me.COCOM_List, "[ID_COCOM]") &
        AND " &
        MultiListItems(Me.Branch_List, "[ID_ServiceBranch]") & GetCurrentUserFilter(False,
        False) &
        ORDER BY Country, Location"
        Me.Branch_List.Enabled = False

        Me.Location_List.Enabled = False
        Me.All_Locations.Value = True

    Else
        Me.Branch_List.Enabled = True
    End If
End Sub

```

- After All Billes Checkbox Update

```

Private Sub All_Billes_AfterUpdate()
'Make Billes List unselectable when box for All is checked
If Me.All_Billes Then
    Dim varItem As Variant
    For Each varItem In Me.Billet_List.ItemsSelected
        Me.Billet_List.Selected(varItem) = False
    Next varItem
    Me.Billet_List.Enabled = False

Else
    Me.Billet_List.Enabled = True
End If
End Sub

```

- Branch List Click

```

Private Sub Branch_List_Click()
    Me.Location_List.RowSource = "SELECT DISTINCT ID_Location, Location, Country FROM
    qryLocationListFiltered WHERE " & MultiListItems(Me.COCOM_List, "[ID_COCOM]") &
    AND " &
    MultiListItems(Me.Branch_List, "[ID_ServiceBranch]") & GetCurrentUserFilter(False,
    False) &
    ORDER BY Country, Location"
    Me.Location_List.Enabled = False
    Me.All_Locations.Value = True
End Sub

```

- COCOM List Click

```

Private Sub COCOM_List_Click()
    Me.Branch_List.RowSource = "SELECT DISTINCT ID_ServiceBranch, ServiceBranch FROM
    qryBranchListFiltered WHERE " & MultiListItems(Me.COCOM_List, "[ID_COCOM]") &
    GetCurrentUserFilter(False, False)
    Me.Location_List.RowSource = "SELECT DISTINCT ID_Location, Location, Country FROM
    qryLocationListFiltered WHERE " & MultiListItems(Me.COCOM_List, "[ID_COCOM]") &
    AND " &
    MultiListItems(Me.Branch_List, "[ID_ServiceBranch]") & GetCurrentUserFilter(False,
    False) &
    ORDER BY Country, Location"
    Me.Branch_List.Enabled = False
    Me.All_Branchs.Value = True
    Me.Location_List.Enabled = False
    Me.All_Locations.Value = True
End Sub

```

- On Open

```

Private Sub Form_Open(Cancel As Integer)
' Initialize the form with the open arguments.
    Me.Filter = "[ReportID] = " & Nz(Me.OpenArgs, 1)
    Me.FilterOn = True
    Me.Label1.Caption = Me![ReportTitle]

```

```

Me.COCOM_List.RowSource = "SELECT COCOMs.* FROM COCOMs " &
GetCurrentUserFilter(True, True)
Me.Branch_List.RowSource = "SELECT DISTINCT ID_ServiceBranch, ServiceBranch FROM
qryBranchListFiltered WHERE " & MultiListItems(Me.COCOM_List, "[ID_COCOM]") &
GetCurrentUserFilter(False, False)
Me.Location_List.RowSource = "SELECT DISTINCT ID_Location, Location, Country FROM
qryLocationListFiltered WHERE " & MultiListItems(Me.COCOM_List, "[ID_COCOM]") &
"AND " &
MultiListItems(Me.Branch_List, "[ID_ServiceBranch]") & GetCurrentUserFilter(False,
False) & "
ORDER BY Country, Location"

Me.COCOM_List.Enabled = False
Me.Location_List.Enabled = False
Me.Branch_List.Enabled = False
Me.SearchFieldLabel.Caption = IIf(IsNull(Me![SearchFieldName]), "", ,
Me![SearchFieldName])
Me.SearchFieldBox.Visible = Me![SearchField]
Me.SearchFieldLabel.Visible = Me![SearchField]
Me.All_Billets.Visible = Me![Billets]
Me.All_BilletsLabel.Visible = Me![Billets]
Me.BilletsLabel.Visible = Me![Billets]
Me.Billet_List.Visible = Me![Billets]
End Sub

```

- OK Button Click

```

Private Sub OK_Click()
On Error GoTo Err_OK_Click
Dim strFilter As String
Dim strReportName As String

strReportName = Me![ReportName]
strFilter = MultiListItems(Me.COCOM_List, "[ID_COCOM]")
strFilter = strFilter & " AND " & MultiListItems(Me.Location_List,
"[ID_Location]")
strFilter = strFilter & " AND " & MultiListItems(Me.Branch_List,
"[ID_ServiceBranch]")
strFilter = strFilter & GetCurrentUserFilter(False, False)

If Me.Billets Then strFilter = strFilter & " AND " &
MultiListItems(Me.Billet_List, "[Billet]")

If Me.SearchField Then strFilter = strFilter & BuildSearchFieldQuerey

Debug.Print strFilter

DoCmd.OpenReport strReportName, acPreview, , strFilter
DoCmd.Close acForm, "frmOpenReports"
Exit_OK_Click:
Exit Sub
Err_OK_Click:
MsgBox Err.Description
Resume Exit_OK_Click
End Sub

```

- Cancel Button Click

```

Private Sub Cancel_Click()
On Error GoTo Err_Cancel_Click
DoCmd.Close
Exit_Cancel_Click:
Exit Sub
Err_Cancel_Click:
MsgBox Err.Description
Resume Exit_Cancel_Click
End Sub

```

- Private Function Multi List Items To String

```

Private Function MultiListItemsToString(targetList As ListBox) As String
    Dim strFilter As String
    Dim varItem As Variant
    strFilter = "("
    For Each varItem In targetList.ItemsSelected
        strFilter = strFilter & targetList.ItemData(varItem) & ","
    Next varItem
    strFilter = Left$(strFilter, Len(strFilter) - 1) & ")"
    MultiListItemsToString = strFilter
End Function

```

- Private Function Multi List Items

```

Private Function MultiListItems(targetList As ListBox, targetField As String) As String
    Dim count As Integer
    Dim varItem As Variant
    count = 0
    For Each varItem In targetList.ItemsSelected
        count = count + 1
    Next varItem
    If count > 0 Then
        MultiListItems = targetField & " IN " & MultiListItemsToString(targetList)
    Else
        MultiListItems = " True "
    End If
End Function

```

- Private Function Build Search Field Query

```

Private Function BuildSearchFieldQuerey() As String
    Dim strFilter As String
    If Me.SearchField And Not IsNull(Me.SearchFieldBox.Value) Then
        strFilter = " And (" & Replace(Me!SearchClause, "$$", Me.SearchFieldBox.Value) & ")"
    Else
        strFilter = ""
    End If
    BuildSearchFieldQuerey = strFilter
End Function

```

## Controls

- All Billets: Unbound
- All Branchs: Unbound
- All COCOMs: Unbound
- All Locations: Unbound
- Billets: Unbound
- Branchs: Unbound
- COCOMs: Unbound
- Locations: Unbound
- Search Field: Unbound
- OK(Button): btnOK\_Click
- Cancel(Button): btnCancel\_Click

## I. frmTargetID

### Controls

- TargetAuditTrail(Unlabeled): Unbound
- TargetID(Unlabeled): Unbound

## J. Locations

### Record Source

- Locations

### Order By

- Country
- Location\_State

### Controls

- CONUS: CONUS
- Country: Country
- Location: Location\_State

## K. PendingDatasheetView

### Record Source

- qryPendingSiteFormFilter

### Dependencies

- Drop-Down Lists
  - Command Name: qryCommandList
  - COCOM: COCOMs
  - Location: qryLocationList
  - Service Branch: ServiceBranchs
  - ProjMgr: ProjectManagers

### Events

- Before Update

```
Private Sub Form_BeforeUpdate(Cancel As Integer)
    Call Audit_Trail(Me)
End Sub
```

- On Delete

```
Private Sub Form_Delete(Cancel As Integer)
    Call Audit_Trail_Delete(Me)
    Dim stDocName As String
    DoCmd.OpenForm "frmTargetID", acNormal, , , acFormEdit, acWindowNormal
    Forms!frmTargetID.TargetID = Me.ID_Auto
    Forms!frmTargetID.TargetAuditTrail = Me.AuditTrail
    DoCmd.SetWarnings (False)
    stDocName = "qryBackupDeletedSite"
    DoCmd.OpenQuery stDocName, acViewNormal, acEdit
    stDocName = "qryBackupDeletedSiteSurvey"
    DoCmd.OpenQuery stDocName, acViewNormal, acEdit
    stDocName = "qryBackupDeletedMaint"
    DoCmd.OpenQuery stDocName, acViewNormal, acEdit
    stDocName = "qryBackupDeletedFunding"
    DoCmd.OpenQuery stDocName, acViewNormal, acEdit
    stDocName = "qryBackupDeletedActionItemLinks"
    DoCmd.OpenQuery stDocName, acViewNormal, acEdit
    stDocName = "qryBackupDeletedContactLinks"
    DoCmd.OpenQuery stDocName, acViewNormal, acEdit
    DoCmd.SetWarnings (True)
    DoCmd.Close acForm, "frmTargetID", acSaveNo
End Sub
```

- On Open

```

Private Sub Form_Open(Cancel As Integer)
    Me.RecordSource = "SELECT * FROM qryPendingSiteFormFilter " &
    GetCurrentUserFilter(True, False)
End Sub

```

## Controls

- AuditTrail(Hidden): AuditTrail
- City: City
- Command Name: CommandName
- Config Training Date: ConfigTrainingDate
- Config Training Duration: ConfigTrainingDuration
- Config Training Team: ConfigTrainingTeam
- Current Code Base: CurrentCodeBase
- Current Handheld Version Number: CurrentHandHeldVersionNumber
- Current Handheld Server Version Number: CurrentHHSVersionNumber
- CurrentServerLocationSite: CurrentServerLocationSite
- CurrentSystemVersionNumber: CurrentSystemVersionNumber
- DatabaseVersion: DatabaseVersion
- DBIDSCards: DBIDSCards
- DEERSPreLoadDate: DEERSPreLoadDate
- DeployDate: DeployDate
- DIPMSTELOSEntryDate: DIPMSTELOSEntryDate
- EquipBuildTELOSDuration: EquipBuildTELOSDuration
- EquipBuildTELOSStartDate: EquipBuildTELOSStartDate
- Equip Order Date: EquipOrderDate
- EquipReceivedSiteDate: EquipReceivedSiteDate
- EquipReceivedTELOSDate: EquipReceivedTELOSDate
- ExtraHandheldsInstalled: ExtraHandheldsInstalled
- ExtraHandheldsPlanned: ExtraHandheldsPlanned
- FieldingTeam: FieldingTeam
- GatesInstalled: GatesInstalled
- GatesPlanned: GatesPlanned
- GateStationInstallDate: GateStationInstallDate
- GateStationInstallDuration: GateStationInstallDuration
- COCOM: ID\_COCOM
- Location: ID\_Location
- Service Branch: ID\_ServiceBranch
- ID\_Site: ID\_Site
- LEOInstalled: LEOInstalled
- LEOPlanned: LEOPlanned
- MaintStartDate: MaintStartDate
- MaintYN: MaintYN
- ManPowerEffecientGateInstalled: ManPowerEffecientGateInstalled
- ManPowerEffecientGatePlanned: ManPowerEffecientGatePlanned
- MobileKitsInstalled: MobileKitsInstalled
- MobileKitsPlanned: MobileKitsPlanned
- NextCodeBase: NextCodeBase
- NextHandHeldVersionNumber: NextHandHeldVersionNumber
- NextHHSVersionNumber: NextHHSVersionNumber
- NextServerLocationSite: NextServerLocationSite
- NextSystemVersionNumber: NextSystemVersionNumber
- Notes: Notes
- Number: Number

- OracleLicenseVersion: OracleLicenseVersion
- ProjectID: ProjectID
- ProjMgr: ProjMgr
- RegistrationInstalled: RegistrationInstalled
- RegistrationPlanned: RegistrationPlanned
- RegStationInstallDate: RegStationInstallDate
- RegStationInstallDuration: RegStationInstallDuration
- ServerOS: ServerOS
- SiteIDAssigned: SiteIDAssigned
- SiteLOCDate: SiteLOCDate
- SiteName: SiteName
- SiteSSAADate: SiteSSAADate
- SiteSSAALead: SiteSSAALead
- SSMChecklistDate: SSMChecklistDate
- SvcMOUReceivedDate: SvcMOUReceivedDate
- TotalArchivedCurrent: TotalArchivedCurrent
- TotalCurrent: TotalCurrent
- Vendors: Vendors
- VisitorCenterInstalled: VisitorCenterInstalled
- VisitorCenterPlanned: VisitorCenterPlanned
- VPNChecklistDate: VPNChecklistDate
- VPNConnectDate: VPNConnectDate

## L. PendingSites

### Record Source

- qryPendingSiteFormFilter

### Dependencies

- Drop-Down Lists
  - Parent Site: qryParentSites
  - Command Name: qryCommandList
  - COCOM: COCOMs
  - Location: qryLocationList
  - Service Branch: ServiceBranchs
  - ProjMgr: ProjectManagers
- Subforms
  - Maintenance: subformMaint
  - Action Items: subformActionItems
  - Funding: subformFunding
  - Site Survey: subformSurveys

### Events

- Private Sub Show Parent Site

```

Private Sub Show_Parent_Site()
On Error GoTo Error_exit
  If Me.PrimarySite.Value Then
    Me.ID_ParentSite.Visible = False
    Me.ID_ParentSite.Value = Null
  Else
    Me.ID_ParentSite.Visible = True
  End If
Error_exit:
End Sub

```

- Add New Branch Click

```
Private Sub AddNewBranch_Click()
    DoCmd.OpenForm "ServiceBranchs", acNormal, , , acDialog
End Sub
```

- Add New COCOM Click

```
Private Sub AddNewCOCOM_Click()
    DoCmd.OpenForm "COCOMs", acNormal, , , acDialog
End Sub
```

- Add New Location Click

```
Private Sub AddNewLocation_Click()
    DoCmd.OpenForm "Locations", acNormal, , , acDialog
End Sub
```

- Add New Manager Click

```
Private Sub AddNewManager_Click()
    DoCmd.OpenForm "ProjectManagers", acNormal, , , acDialog
End Sub
```

- Clear Button Click

```
Private Sub btnClear_Click()
    Me.Search_Field.Value = ""
    Me.Filter = ""
    Me.FilterOn = False
End Sub
```

- Search Name Button Click

```
Private Sub btnSearchName_Click()
    Me.Filter = "SiteName Like '*' & Me.Search_Field & '*'"
    Me.FilterOn = True
    If Me.Recordset.RecordCount = 0 Then
        Me.Filter = ""
        Me.FilterOn = True
        MsgBox "There are no records that meet that criteria!!"
    End If
End Sub
```

- On Delete

```
Private Sub Form_Delete(Cancel As Integer)
    Call Audit_Trail_Delete(Me)
    Dim stDocName As String
    DoCmd.OpenForm "frmTargetID", acNormal, , , acFormEdit, acWindowNormal

    Forms!frmTargetID.TargetID = Me.ID_Auto
    Forms!frmTargetID.TargetAuditTrail = Me.AuditTrail

    DoCmd.SetWarnings (False)

    stDocName = "qryBackupDeletedSite"
    DoCmd.OpenQuery stDocName, acViewNormal, acEdit

    stDocName = "qryBackupDeletedSiteSurvey"
    DoCmd.OpenQuery stDocName, acViewNormal, acEdit

    stDocName = "qryBackupDeletedMaint"
    DoCmd.OpenQuery stDocName, acViewNormal, acEdit

    stDocName = "qryBackupDeletedFunding"
    DoCmd.OpenQuery stDocName, acViewNormal, acEdit

    stDocName = "qryBackupDeletedActionItemLinks"
    DoCmd.OpenQuery stDocName, acViewNormal, acEdit
```

```

stDocName = "qryBackupDeletedContactLinks"
DoCmd.OpenQuery stDocName, acViewNormal, acEdit

DoCmd.SetWarnings (True)
DoCmd.Close acForm, "frmTargetID", acSaveNo
End Sub

```

- After Goto List Update

```

Private Sub GotoList_AfterUpdate()
    Dim rs As Object
    Set rs = Me.Recordset.Clone
    rs.FindFirst "[ID_Auto] = " & Me![GotoList]
    Me.Bookmark = rs.Bookmark
End Sub

```

- Before Update

```

Private Sub Form_BeforeUpdate(Cancel As Integer)
    If Not CurrentUser = "DBIDS Manager" Then Call Audit_Trail(Me)
End Sub

```

- On Current

```

Private Sub Form_Current()
    Show_Parent_Site
    Me.Refresh
End Sub

```

- On Open

```

Private Sub Form_Open(Cancel As Integer)
    If (IsNull(Me.OpenArgs)) Then
        Dim strFilter As String
        strFilter = GetCurrentUserFilter(True, False)
        Me.RecordSource = "SELECT * FROM qryPendingSiteFormFilter " & strFilter
        If strFilter = "" Then
            Me.GotoList.RowSource = "SELECT * FROM qrySiteID_Name_Location " &
" WHERE ID_Site = 'None'"
        Else
            Me.GotoList.RowSource = "SELECT * FROM qrySiteID_Name_Location " &
strFilter & " AND
ID_Site = 'None'"
        End If
    End If
End Sub

```

- After Primary Site Update

```

Private Sub PrimarySite_AfterUpdate()
    Show_Parent_Site
End Sub

```

## Controls

- Goto Site: Unbound
- Search Field(Unlabeled): Unbound
- Search Name(Button): btnSearchName
- Clear(Button): btnClear
- Site ID: ID\_Site
- SiteName: SiteName
- Is a Primary Site: PrimarySite
- Parent Site: ID\_ParentSite
- AuditTrail(Hidden): AuditTrail
- City: City
- Location: ID\_Location
- Add New Location(Button): AddNewLocation\_Click

- Command Name: CommandName
- COCOM: ID\_COCOM
- Add New COCOM(Button): AddNewCOCOM\_Click
- Service Branch: ID\_ServiceBranch
- Add New Branch(Button): AddNewBranch\_Click
- Administrative Actions / Integration(Tab)
- SvcMOUReceivedDate: SvcMOUReceivedDate
- SSMChecklistDate: SSMChecklistDate
- VPNChecklistDate: VPNChecklistDate
- SiteIDAssigned: SiteIDAssigned
- SiteSSAADate: SiteSSAADate
- SiteSSAALead: SiteSSAALead
- Number: Number
- Vendors: Vendors
- Equipment Processing / Fielding(Tab)
- Equip Order Date: EquipOrderDate
- EquipReceivedTELOSDate: EquipReceivedTELOSDate
- EquipBuildTELOSStartDate: EquipBuildTELOSStartDate
- EquipBuildTELOSDuration: EquipBuildTELOSDuration
- EquipReceivedSiteDate: EquipReceivedSiteDate
- DIPMSTELOSEntryDate: DIPMSTELOSEntryDate
- RegStationInstallDate: RegStationInstallDate
- RegStationInstallDuration: RegStationInstallDuration
- GateStationInstallDate: GateStationInstallDate
- GateStationInstallDuration: GateStationInstallDuration
- Config Training Date: ConfigTrainingDate
- Config Training Duration: ConfigTrainingDuration
- FieldingTeam: FieldingTeam
- VPNConnectDate: VPNConnectDate
- DEERSPreLoadDate: DEERSPreLoadDate
- Config Training Team: ConfigTrainingTeam
- SiteLOCDate: SiteLOCDate
- Funding / Action Items(Tab)
- Refresh / Maintenance(Tab)
- Site Survey(Tab)
- Version/Server(Tab)
- Current Code Base: CurrentCodeBase
- Current Handheld Version Number: CurrentHandHeldVersionNumber
- Current Handheld Server Version Number: CurrentHHServerVersionNumber
- CurrentServerLocationSite: CurrentServerLocationSite
- CurrentSystemVersionNumber: CurrentSystemVersionNumber
- NextCodeBase: NextCodeBase
- NextHandHeldVersionNumber: NextHandHeldVersionNumber
- NextHHServerVersionNumber: NextHHServerVersionNumber
- NextServerLocationSite: NextServerLocationSite
- NextSystemVersionNumber: NextSystemVersionNumber
- ServerOS: ServerOS
- DatabaseVersion: DatabaseVersion
- OracleLicenseVersion: OracleLicenseVersion
- DeployDate: DeployDate
- Work Stations / Registrants / Status(Tab)
- RegistrationInstalled: RegistrationInstalled

- RegistrationPlanned: RegistrationPlanned
- LEOInstalled: LEOInstalled
- LEOPlanned: LEOPlanned
- GatesInstalled: GatesInstalled
- GatesPlanned: GatesPlanned
- ManPowerEffecientGateInstalled: ManPowerEffecientGateInstalled
- ManPowerEffecientGatePlanned: ManPowerEffecientGatePlanned
- MobileKitsInstalled: MobileKitsInstalled
- MobileKitsPlanned: MobileKitsPlanned
- VisitorCenterInstalled: VisitorCenterInstalled
- VisitorCenterPlanned: VisitorCenterPlanned
- ExtraHandheldsInstalled: ExtraHandheldsInstalled
- ExtraHandheldsPlanned: ExtraHandheldsPlanned
- Status%: =IIf(([Registration Installed]+[LEO Installed]+[Gates Installed]+[Man Power Effecient Gate Installed]+[Mobile Kits Installed]+[Visitor Center Installed]+[Extra Handhelds Installed])=0,0,([Registration Installed]+[LEO Installed]+[Gates Installed]+[Man Power Effecient Gate Installed]+[Mobile Kits Installed]+[Visitor Center Installed]+[Extra Handhelds Installed])/([Registration Planned]+[LEO Planned]+[Gates Planned]+[Man Power Effecient Gate Planned]+[Mobile Kits Planned]+[Visitor Center Planned]+[Extra Handhelds Planned]))
- TotalCurrent: TotalCurrent
- TotalArchivedCurrent: TotalArchivedCurrent
- DBIDSCards: DBIDSCards
- MaintYN: MaintYN
- MaintStartDate: MaintStartDate
- ProjMgr: ProjMgr
- Add New Manager(Button): AddNewManager\_Click
- Notes: Notes
- ProjectID: ProjectID

## M. PlannedInstalledErrors

### Record Source

- qryPlannedInstalledErrors

### Events

- Edit Button Click

```

Private Sub Edit_Click()
On Error GoTo Err_Edit_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    If Me![ID_Site] = "None" Then
        stDocName = "PendingSites"
    Else
        stDocName = "Sites"
    End If

    stLinkCriteria = "[ID_Auto] = " & Me![ID_Auto]
    Debug.Print stLinkCriteria
    Debug.Print stDocName
    DoCmd.OpenForm stDocName, , , stLinkCriteria, , stLinkCriteria

Exit_Edit_Click:
    Exit Sub
Err_Edit_Click:
    MsgBox Err.Description

```

*Resume* *Exit>Edit\_Click*  
*End Sub*

### **Controls**

- COCOM
- ID\_Auto(Hidden): ID\_Auto
- Site ID: ID\_Site
- Site Name: SiteName
- Service Branch: ServiceBranch
- RegistrationInstalled: RegistrationInstalled
- RegistrationPlanned: RegistrationPlanned
- LEOInstalled: LEOInstalled
- LEOPlanned: LEOPlanned
- GatesInstalled: GatesInstalled
- GatesPlanned: GatesPlanned
- ManPowerEffecientGateInstalled: ManPowerEffecientGateInstalled
- ManPowerEffecientGatePlanned: ManPowerEffecientGatePlanned
- MobileKitsInstalled: MobileKitsInstalled
- MobileKitsPlanned: MobileKitsPlanned
- VisitorCenterInstalled: VisitorCenterInstalled
- VisitorCenterPlanned: VisitorCenterPlanned
- ExtraHandheldsInstalled: ExtraHandheldsInstalled
- ExtraHandheldsPlanned: ExtraHandheldsPlanned
- Edit(Button): Edit\_Click

## **N. ProjectManagers**

### **Record Source**

- ProjectManagers

### **Controls**

- Project Manager Name: ProjectManagerName

## **O. ServiceBranchs**

### **Record Source**

- ServiceBranchs

### **Controls**

- Service Branch: SearchBranch

## **P. SiteBackup**

### **Record Source**

- Sites\_Backup

### **Dependencies**

- Drop-Down Lists
  - Parent Site: qryParentSites
  - Command Name: qryCommandList
  - COCOM: COCOMs

- Location: qryLocationList
- Service Branch: ServiceBranchs
- ProjMgr: ProjectManagers
- Subforms
  - Maintenance: subformBackupMaint
  - Funding: subformBackupFunding
  - Site Survey: subformBackupSurveys

## Events

- On Current

```
Private Sub Form_Current()
  If Me.Recordset.EOF Then
    DoCmd.Close acForm, "SiteBackup", acSaveNo
  End If
End Sub
```

- On Open

```
Private Sub Form_Open(Cancel As Integer)
  If Me.Recordset.EOF Then
    Me.RestoreSiteBtn.Enabled = False
  Else
    Me.RestoreSiteBtn.Enabled = True
  End If
End Sub
```

- Restore Site Button Click

```
Private Sub RestoreSiteBtn_Click()
On Error GoTo Err_RestoreSiteBtn

  Dim stDocName As String
  DoCmd.SetWarnings (False)

  DoCmd.OpenForm "frmTargetID", acNormal, , , acFormEdit, acWindowNormal

  Forms!frmTargetID.TargetID = Me.ID_Auto
  Forms!frmTargetID.TargetAuditTrail = Me.AuditTrail

  stDocName = "qryRestoreDeletedSite"
  DoCmd.OpenQuery stDocName, acViewNormal, acEdit

  '*****Get the new ID_Auto

  Dim con As Object
  Dim rs As Object
  Dim stSql As String

  ' Open the table of Switchboard Items, and find
  ' the first item for this Switchboard Page.
  Set con = Application.CurrentProject.Connection
  stSql = "SELECT ID_Auto FROM [Sites] WHERE [AuditTrail] = '***RESTORING***'"
  Set rs = CreateObject("ADoDB.Recordset")
  rs.Open stSql, con, 1
  Forms!frmTargetID.TargetID = rs![ID_Auto]
  rs.Close
  Set rs = Nothing
  Set con = Nothing
  '***** *****

  stDocName = "qryRestoreDeletedSiteSurvey"
  DoCmd.OpenQuery stDocName, acViewNormal, acEdit

  stDocName = "qryRestoreDeletedMaint"
  DoCmd.OpenQuery stDocName, acViewNormal, acEdit
```

```

stDocName = "qryRestoreDeletedFunding"
DoCmd.OpenQuery stDocName, acViewNormal, acEdit

' *****post the audit Trail
stDocName = "qryRestoringSiteAuditTrail"
DoCmd.OpenQuery stDocName, acViewNormal, acEdit
' *****

stDocName = "qryDeleteRestoredBackup"
DoCmd.OpenQuery stDocName, acViewNormal, acEdit

DoCmd.Close acForm, "frmTargetID", acSaveNo

DoCmd.SetWarnings (True)
Me.Requery
Err_RestoreSiteBtn:
End Sub

```

## Controls

- RestoreSiteBtn(Button): RestoreSiteBtn\_Click
- Site ID: ID\_Site
- SiteName: SiteName
- Is a Primary Site: PrimarySite
- Parent Site: ID\_ParentSite
- AuditTrail(Hidden): AuditTrail
- City: City
- Location: ID\_Location
- Command Name: CommandName
- COCOM: ID\_COCOM
- Service Branch: ID\_ServiceBranch
- Administrative Actions / Integration(Tab)
- SvcMOUReceivedDate: SvcMOUReceivedDate
- SSMChecklistDate: SSMChecklistDate
- VPNChecklistDate: VPNChecklistDate
- SiteIDAssigned: SiteIDAssigned
- SiteSSAADate: SiteSSAADate
- SiteSSAALead: SiteSSAALead
- Number: Number
- Vendors: Vendors
- Equipment Processing / Fielding(Tab)
- Equip Order Date: EquipOrderDate
- EquipReceivedTELOSDate: EquipReceivedTELOSDate
- EquipBuildTELOSStartDate: EquipBuildTELOSStartDate
- EquipBuildTELOSDuration: EquipBuildTELOSDuration
- EquipReceivedSiteDate: EquipReceivedSiteDate
- DIPMSTELOSEntryDate: DIPMSTELOSEntryDate
- RegStationInstallDate: RegStationInstallDate
- RegStationInstallDuration: RegStationInstallDuration
- GateStationInstallDate: GateStationInstallDate
- GateStationInstallDuration: GateStationInstallDuration
- Config Training Date: ConfigTrainingDate
- Config Training Duration: ConfigTrainingDuration
- FieldingTeam: FieldingTeam
- VPNConnectDate: VPNConnectDate
- DEERSPreLoadDate: DEERSPreLoadDate
- Config Training Team: ConfigTrainingTeam

- SiteLOCDate: SiteLOCDate
- Funding / Action Items(Tab)
- Refresh / Maintenance(Tab)
- Site Survey(Tab)
- Version/Server(Tab)
- Current Code Base: CurrentCodeBase
- Current Handheld Version Number: CurrentHandHeldVersionNumber
- Current Handheld Server Version Number: CurrentHHSERVERVersionNumber
- CurrentServerLocationSite: CurrentServerLocationSite
- CurrentSystemVersionNumber: CurrentSystemVersionNumber
- NextCodeBase: NextCodeBase
- NextHandHeldVersionNumber: NextHandHeldVersionNumber
- NextHHSERVERVersionNumber: NextHHSERVERVersionNumber
- NextServerLocationSite: NextServerLocationSite
- NextSystemVersionNumber: NextSystemVersionNumber
- ServerOS: ServerOS
- DatabaseVersion: DatabaseVersion
- OracleLicenseVersion: OracleLicenseVersion
- DeployDate: DeployDate
- Work Stations / Registrants / Status(Tab)
- RegistrationInstalled: RegistrationInstalled
- RegistrationPlanned: RegistrationPlanned
- LEOInstalled: LEOInstalled
- LEOPlanned: LEOPlanned
- GatesInstalled: GatesInstalled
- GatesPlanned: GatesPlanned
- ManPowerEffecientGateInstalled: ManPowerEffecientGateInstalled
- ManPowerEffecientGatePlanned: ManPowerEffecientGatePlanned
- MobileKitsInstalled: MobileKitsInstalled
- MobileKitsPlanned: MobileKitsPlanned
- VisitorCenterInstalled: VisitorCenterInstalled
- VisitorCenterPlanned: VisitorCenterPlanned
- ExtraHandheldsInstalled: ExtraHandheldsInstalled
- ExtraHandheldsPlanned: ExtraHandheldsPlanned
- Status%: =IIf(([Registration Installed]+[LEO Installed]+[Gates Installed]+[Man Power Effecient Gate Installed]+[Mobile Kits Installed]+[Visitor Center Installed]+[Extra Handhelds Installed])=0,0,([Registration Installed]+[LEO Installed]+[Gates Installed]+[Man Power Effecient Gate Installed]+[Mobile Kits Installed]+[Visitor Center Installed]+[Extra Handhelds Installed])/([Registration Planned]+[LEO Planned]+[Gates Planned]+[Man Power Effecient Gate Planned]+[Mobile Kits Planned]+[Visitor Center Planned]+[Extra Handhelds Planned]))
- TotalCurrent: TotalCurrent
- TotalArchivedCurrent: TotalArchivedCurrent
- DBIDSCards: DBIDSCards
- MaintYN: MaintYN
- MaintStartDate: MaintStartDate
- ProjMgr: ProjMgr
- Notes: Notes

## Q. SiteDatasheetView

### Record Source

- qrySiteFormFilter

### Dependencies

- Drop-Down Lists
  - Command Name: qryCommandList
  - COCOM: COCOMs
  - Location: qryLocationList
  - Service Branch: ServiceBranchs
  - ProjMgr: ProjectManagers

### Events

- Before Update

```
Private Sub Form_BeforeUpdate(Cancel As Integer)
    Call Audit_Trail(Me)
End Sub
```

- On Delete

```
Private Sub Form_Delete(Cancel As Integer)
    Call Audit_Trail_Delete(Me)
    Dim stDocName As String
    DoCmd.OpenForm "frmTargetID", acNormal, , , acFormEdit, acWindowNormal
    Forms!frmTargetID.TargetID = Me.ID_Auto
    Forms!frmTargetID.TargetAuditTrail = Me.AuditTrail
    DoCmd.SetWarnings (False)
    stDocName = "qryBackupDeletedSite"
    DoCmd.OpenQuery stDocName, acViewNormal, acEdit
    stDocName = "qryBackupDeletedSiteSurvey"
    DoCmd.OpenQuery stDocName, acViewNormal, acEdit
    stDocName = "qryBackupDeletedMaint"
    DoCmd.OpenQuery stDocName, acViewNormal, acEdit
    stDocName = "qryBackupDeletedFunding"
    DoCmd.OpenQuery stDocName, acViewNormal, acEdit
    stDocName = "qryBackupDeletedActionItemLinks"
    DoCmd.OpenQuery stDocName, acViewNormal, acEdit
    stDocName = "qryBackupDeletedContactLinks"
    DoCmd.OpenQuery stDocName, acViewNormal, acEdit
    DoCmd.SetWarnings (True)
    DoCmd.Close acForm, "frmTargetID", acSaveNo
End Sub
```

- On Open

```
Private Sub Form_Open(Cancel As Integer)
    Me.RecordSource = "SELECT * FROM qrySiteFormFilter " &
        GetCurrentUserFilter(True, False)
End Sub
```

### Controls

- AuditTrail(Hidden): AuditTrail
- City: City
- Command Name: CommandName
- Config Training Date: ConfigTrainingDate
- Config Training Duration: ConfigTrainingDuration
- Config Training Team: ConfigTrainingTeam
- Current Code Base: CurrentCodeBase
- Current Handheld Version Number: CurrentHandHeldVersionNumber
- Current Handheld Server Version Number: CurrentHHSERVERVersionNumber

- CurrentServerLocationSite: CurrentServerLocationSite
- CurrentSystemVersionNumber: CurrentSystemVersionNumber
- DatabaseVersion: DatabaseVersion
- DBIDSCards: DBIDSCards
- DEERSPreLoadDate: DEERSPreLoadDate
- DeployDate: DeployDate
- DIPMSTELOSEntryDate: DIPMSTELOSEntryDate
- EquipBuildTELOSDuration: EquipBuildTELOSDuration
- EquipBuildTELOSStartDate: EquipBuildTELOSStartDate
- Equip Order Date: EquipOrderDate
- EquipReceivedSiteDate: EquipReceivedSiteDate
- EquipReceivedTELOSDate: EquipReceivedTELOSDate
- ExtraHandheldsInstalled: ExtraHandheldsInstalled
- ExtraHandheldsPlanned: ExtraHandheldsPlanned
- FieldingTeam: FieldingTeam
- GatesInstalled: GatesInstalled
- GatesPlanned: GatesPlanned
- GateStationInstallDate: GateStationInstallDate
- GateStationInstallDuration: GateStationInstallDuration
- COCOM: ID\_COCOM
- Location: ID\_Location
- Service Branch: ID\_ServiceBranch
- ID\_Site: ID\_Site
- LEOInstalled: LEOInstalled
- LEOPlanned: LEOPlanned
- MaintStartDate: MaintStartDate
- MaintYN: MaintYN
- ManPowerEffecientGateInstalled: ManPowerEffecientGateInstalled
- ManPowerEffecientGatePlanned: ManPowerEffecientGatePlanned
- MobileKitsInstalled: MobileKitsInstalled
- MobileKitsPlanned: MobileKitsPlanned
- NextCodeBase: NextCodeBase
- NextHandHeldVersionNumber: : NextHandHeldVersionNumber
- NextHHSERVERVersionNumber: NextHHSERVERVersionNumber
- NextServerLocationSite: NextServerLocationSite
- NextSystemVersionNumber: NextSystemVersionNumber
- Notes: Notes
- Number: Number
- OracleLicenseVersion: OracleLicenseVersion
- ProjectID: ProjectID
- ProjMgr: ProjMgr
- RegistrationInstalled: RegistrationInstalled
- RegistrationPlanned: RegistrationPlanned
- RegStationInstallDate: RegStationInstallDate
- RegStationInstallDuration: RegStationInstallDuration
- ServerOS: ServerOS
- SiteIDAssigned: SiteIDAssigned
- SiteLOCDate: SiteLOCDate
- SiteName: SiteName
- SiteSSAADate: SiteSSAADate
- SiteSSAALead: SiteSSAALead
- SSMChecklistDate: SSMChecklistDate

- SvcMOUReceivedDate: SvcMOUReceivedDate
- TotalArchivedCurrent: TotalArchivedCurrent
- TotalCurrent: TotalCurrent
- Vendors: Vendors
- VisitorCenterInstalled: VisitorCenterInstalled
- VisitorCenterPlanned: VisitorCenterPlanned
- VPNChecklistDate: VPNChecklistDate
- VPNConnectDate: VPNConnectDate

## R. Sites

### Record Source

- qrySiteFormFilter

### Dependencies

- Drop-Down Lists
  - Parent Site: qryParentSites
  - Command Name: qryCommandList
  - COCOM: COCOMs
  - Location: qryLocationList
  - Service Branch: ServiceBranchs
  - ProjMgr: ProjectManagers
- Subforms
  - Maintenance: subformMaint
  - Action Items: subformActionItems
  - Funding: subformFunding
  - Site Survey: subformSurveys

### Events

- Private Sub Show\_Parent\_Site

```
Private Sub Show_Parent_Site()
On Error GoTo Error_exit
    If Me.PrimarySite.Value Then
        Me.ID_ParentSite.Visible = False
        Me.ID_ParentSite.Value = Null
    Else
        Me.ID_ParentSite.Visible = True
    End If
Error_exit:
End Sub
```

- Add New Branch Button Click

```
Private Sub AddNewBranch_Click()
    DoCmd.OpenForm "ServiceBranchs", acNormal, , , , acDialog
End Sub
```

- Add New COCOM Button Click

```
Private Sub AddNewCOCOM_Click()
    DoCmd.OpenForm "COCOMs", acNormal, , , , acDialog
End Sub
```

- Add New Location Button Click

```
Private Sub AddNewLocation_Click()
    DoCmd.OpenForm "Locations", acNormal, , , , acDialog
End Sub
```

- Add New Manager Button Click

```
Private Sub AddNewManager_Click()
  DoCmd.OpenForm "ProjectManagers", acNormal, , , acDialog
End Sub
```

- Clear Button Click

```
Private Sub btnClear_Click()
  Me.Search_Field.Value = ""
  Me.Filter = ""
  Me.FilterOn = False
End Sub
```

- Button Search ID Click

```
Private Sub btnSearchID_Click()
  Me.Filter = "ID_Site Like '*' & Me.Search_Field & '*'"
  Me.FilterOn = True
  If Me.Recordset.RecordCount = 0 Then
    Me.Filter = ""
    Me.FilterOn = True
    MsgBox "There are no records that meet that criteria!!"
  End If
End Sub
```

- Search Name Button Click

```
Private Sub btnSearchName_Click()
  Me.Filter = "SiteName Like '*' & Me.Search_Field & '*'"
  Me.FilterOn = True
  If Me.Recordset.RecordCount = 0 Then
    Me.Filter = ""
    Me.FilterOn = True
    MsgBox "There are no records that meet that criteria!!"
  End If
End Sub
```

- On Delete

```
Private Sub Form_Delete(Cancel As Integer)
  Call Audit_Trail_Delete(Me)
  Dim stDocName As String
  DoCmd.OpenForm "frmTargetID", acNormal, , , acFormEdit, acWindowNormal

  Forms!frmTargetID.TargetID = Me.ID_Auto
  Forms!frmTargetID.TargetAuditTrail = Me.AuditTrail

  DoCmd.SetWarnings (False)

  stDocName = "qryBackupDeletedSite"
  DoCmd.OpenQuery stDocName, acViewNormal, acEdit

  stDocName = "qryBackupDeletedSiteSurvey"
  DoCmd.OpenQuery stDocName, acViewNormal, acEdit

  stDocName = "qryBackupDeletedMaint"
  DoCmd.OpenQuery stDocName, acViewNormal, acEdit

  stDocName = "qryBackupDeletedFunding"
  DoCmd.OpenQuery stDocName, acViewNormal, acEdit

  stDocName = "qryBackupDeletedActionItemLinks"
  DoCmd.OpenQuery stDocName, acViewNormal, acEdit

  stDocName = "qryBackupDeletedContactLinks"
  DoCmd.OpenQuery stDocName, acViewNormal, acEdit

  DoCmd.SetWarnings (True)
  DoCmd.Close acForm, "frmTargetID", acSaveNo
End Sub
```

- After Goto List Update

```
Private Sub GotoList_AfterUpdate()
    Dim rs As Object
    Set rs = Me.Recordset.Clone
    rs.FindFirst "[ID_Auto] = " & Me![GotoList]
    Me.Bookmark = rs.Bookmark
End Sub
```

- Before Update

```
Private Sub Form_BeforeUpdate(Cancel As Integer)
    If Not CurrentUser = "DBIDS Manager" Then Call Audit_Trail(Me)
End Sub
```

- On Current

```
Private Sub Form_Current()
    Show_Parent_Site
    Me.Refresh
End Sub
```

- On Load

```
Private Sub Form_Load()
    If Not IsNull(Forms!Sites.OpenArgs) Then
        strFilter = Forms!Sites.OpenArgs
        DoCmd.FindRecord strFilter, , True, , True, acAll, True
    End If
End Sub
```

- On Open

```
Private Sub Form_Open(Cancel As Integer)
    If (IsNull(Me.OpenArgs)) Then
        Dim strFilter As String
        strFilter = GetCurrentUserFilter(True, False)
        Me.RecordSource = "SELECT * FROM qrySiteFormFilter " & strFilter
        If strFilter = "" Then
            Me.GotoList.RowSource = "SELECT * FROM qrySiteID_Name_Location " &
        " WHERE
        ID_Site <> 'None'
        Else
            Me.GotoList.RowSource = "SELECT * FROM qrySiteID_Name_Location " &
        strFilter & " AND
        ID_Site <> 'None'"
        End If
    End If
End Sub
```

- After Primary Site Update

```
Private Sub PrimarySite_AfterUpdate()
    Show_Parent_Site
End Sub
```

## Controls

- Goto Site: Unbound
- Search Name(Unlabeled): Unbound
- Search Name(Button): btnSearchName
- Clear(Button): btnClear
- Site ID: ID\_Site
- SiteName: SiteName
- Is a Primary Site: PrimarySite
- Parent Site: ID\_ParentSite
- AuditTrail(Hidden): AuditTrail

- City: City
- Location: ID\_Location
- Add New Location(Button): AddNewLocation\_Click
- Command Name: CommandName
- COCOM: ID\_COCOM
- Add New COCOM(Button): AddNewCOCOM\_Click
- Service Branch: ID\_ServiceBranch
- Add New Branch(Button): AddNewBranch\_Click
- Administrative Actions / Integration(Tab)
- SvcMOUReceivedDate: SvcMOUReceivedDate
- SSMChecklistDate: SSMChecklistDate
- VPNChecklistDate: VPNChecklistDate
- SiteIDAssigned: SiteIDAssigned
- SiteSSAADate: SiteSSAADate
- SiteSSAALead: SiteSSAALead
- Number: Number
- Vendors: Vendors
- Equipment Processing / Fielding(Tab)
- Equip Order Date: EquipOrderDate
- EquipReceivedTELOSDate: EquipReceivedTELOSDate
- EquipBuildTELOSStartDate: EquipBuildTELOSStartDate
- EquipBuildTELOSDuration: EquipBuildTELOSDuration
- EquipReceivedSiteDate: EquipReceivedSiteDate
- DIPMSTELOSEntryDate: DIPMSTELOSEntryDate
- RegStationInstallDate: RegStationInstallDate
- RegStationInstallDuration: RegStationInstallDuration
- GateStationInstallDate: GateStationInstallDate
- GateStationInstallDuration: GateStationInstallDuration
- Config Training Date: ConfigTrainingDate
- Config Training Duration: ConfigTrainingDuration
- FieldingTeam: FieldingTeam
- VPNConnectDate: VPNConnectDate
- DEERSPreLoadDate: DEERSPreLoadDate
- Config Training Team: ConfigTrainingTeam
- SiteLOCDate: SiteLOCDate
- Funding / Action Items(Tab)
- Refresh / Maintenance(Tab)
- Site Survey(Tab)
- Version/Server(Tab)
- Current Code Base: CurrentCodeBase
- Current Handheld Version Number: CurrentHandHeldVersionNumber
- Current Handheld Server Version Number: CurrentHHSVersionNumber
- CurrentServerLocationSite: CurrentServerLocationSite
- CurrentSystemVersionNumber: CurrentSystemVersionNumber
- NextCodeBase: NextCodeBase
- NextHandHeldVersionNumber: NextHandHeldVersionNumber
- NextHHSVersionNumber: NextHHSVersionNumber
- NextServerLocationSite: NextServerLocationSite
- NextSystemVersionNumber: NextSystemVersionNumber
- ServerOS: ServerOS
- DatabaseVersion: DatabaseVersion
- OracleLicenseVersion: OracleLicenseVersion

- DeployDate: DeployDate
- Work Stations / Registrants / Status(Tab)
- RegistrationInstalled: RegistrationInstalled
- RegistrationPlanned: RegistrationPlanned
- LEOInstalled: LEOInstalled
- LEOPlanned: LEOPlanned
- GatesInstalled: GatesInstalled
- GatesPlanned: GatesPlanned
- ManPowerEffecientGateInstalled: ManPowerEffecientGateInstalled
- ManPowerEffecientGatePlanned: ManPowerEffecientGatePlanned
- MobileKitsInstalled: MobileKitsInstalled
- MobileKitsPlanned: MobileKitsPlanned
- VisitorCenterInstalled: VisitorCenterInstalled
- VisitorCenterPlanned: VisitorCenterPlanned
- ExtraHandheldsInstalled: ExtraHandheldsInstalled
- ExtraHandheldsPlanned: ExtraHandheldsPlanned
- Status%: =IIf(([Registration Installed]+[LEO Installed]+[Gates Installed]+[Man Power Effecient Gate Installed]+[Mobile Kits Installed]+[Visitor Center Installed]+[Extra Handhelds Installed])=0,0,([Registration Installed]+[LEO Installed]+[Gates Installed]+[Man Power Effecient Gate Installed]+[Mobile Kits Installed]+[Visitor Center Installed]+[Extra Handhelds Installed])/([Registration Planned]+[LEO Planned]+[Gates Planned]+[Man Power Effecient Gate Planned]+[Mobile Kits Planned]+[Visitor Center Planned]+[Extra Handhelds Planned]))
- TotalCurrent: TotalCurrent
- TotalArchivedCurrent: TotalArchivedCurrent
- DBIDSCards: DBIDSCards
- MaintYN: MaintYN
- MaintStartDate: MaintStartDate
- ProjMgr: ProjMgr
- Add New Manager(Button): AddNewManager\_Click
- Notes: Notes

## S. SiteShippingAddress

### Record Source

```
SELECT * FROM qryContactsReportBySite WHERE ((qryContactsReportBySite.Billet) = 0);
```

### Order By

- SiteName

### Controls

- Address City: AddressCity
- Address Line 1: AddressLine1
- Address Line 2: AddressLine2
- Address Line 3: AddressLine3
- Address State: AddressState
- Address Zip: Address Zip
- Billet: Billet
- COCOM: COCOM
- Comm Phone #1: CommPhone1
- Comm Phone #2: CommPhone2
- Contact Name: ContactName

- DSN #1: DSN1
- DSN #2: DSN2
- Email: Email
- ID\_Site: ID\_Site
- Location/State: Location
- Service Branch: ServiceBranch
- Ship To Line 1: ShipToLine1
- Ship To Line 2: ShipToLine2
- Ship To Line 3: ShipToLine3
- Ship To City: ShipToCity
- Ship To State: ShipToState
- Ship To Zip: ShipToZip
- Site Name: SiteName

## **T. subformActionItems**

### **Record Source**

- qrySiteActionItems

### **Controls**

- ID\_ActionItem(Hidden): ID\_ActionItem
- ID\_Auto(Hidden): ID\_Auto
- ActionItem: ActionItem
- Notes: Notes

## **U. subformActionItemSites**

### **Record Source**

- ActionItemsToSitesLink

### **Dependencies**

- Drop-Down Lists
  - Site ID/Name: qrySiteID\_Name\_Location

### **Controls**

- ID\_ActionItem(Hidden): ID\_ActionItem
- Site ID/Name: ID\_Auto

## **V. subFormBackupFunding**

### **Record Source**

- Funding\_Backup

### **Controls**

- Date: FundingDate
- ID\_Auto(Hidden): ID\_Auto
- ID\_Funding(Hidden): ID\_Funding
- Notes: Notes
- Reason: Reason
- Received: Received
- Required: Required
- Source: Source

## W. subFormBackupMaint

### Record Source

- Maintenance\_Backup

### Controls

- Date: MaintenanceDate
- ID\_Maintenance(Hidden): ID\_Maintenance
- Notes: Notes
- Type: Type

## X. subFormBackupSurveys

### Record Source

- SiteSurvey\_Backup

### Controls

- Actual Finish Date: ActualFinishDate
- Actual Start Date: ActualStartDate
- Current Status: CurrentStatus
- Delivered Customer: DeliveredCustomer
- Estimated Finish Date: EstimatedFinishDate
- Estimated Start Date: EstimatedStartDate
- ID\_Auto(Hidden): ID\_Auto
- ID\_SiteSurvey(Hidden): ID\_SiteSurvey
- Notes: Notes
- Team: Team
- Report(Unlabeled): Report
- Returned Telos: ReturnedTelos
- Revised Report: RevisedReport
- ROMReceived: ROMReceived
- Initial Review: Initial Review
- Type: Type

## Y. subFormContactInformationToSite

### Record Source

- qryContactToSitesList

### Dependencies

- Drop-Down Lists
  - Site: qrySiteID\_Name\_Location

### Controls

- Billet: Billet
- Site: ID\_Auto
- ID\_Contact(Hidden): ID\_Contact

## Z. subformFunding

### Record Source

- Funding

### **Order By**

- FundingDate

### **Events**

- Before Update

```
Private Sub Form_BeforeUpdate(Cancel As Integer)
    If Not CurrentUser = "DBIDS Manager" Then Call Audit_Trail(Me)
End Sub
```

### **Controls**

- Date: FundingDate
- ID\_Auto(Hidden): ID\_Auto
- ID\_Funding(Hidden): ID\_Funding
- Notes: Notes
- Reason: Reason
- Received: Received
- Required: Required
- Source: Source

## **AA. subFormFundingAudit**

### **Record Source**

- qryFundingAudit

### **Controls**

- Audit Trail: AuditTrail
- Funding Date: FundingDate
- ID\_Auto(Hidden): ID\_Auto

## **BB. subformMaint**

### **Record Source**

- Maintenance

### **Events**

- Before Update

```
Private Sub Form_BeforeUpdate(Cancel As Integer)
    Call Audit_Trail(Me)
End Sub
```

### **Controls**

- Date: MaintenanceDate
- ID\_Maintenance(Hidden): ID\_Maintenance
- Notes: Notes
- Type: Type

## **CC. subFormMaintenanceAudit**

### **Record Source**

- qryMaintenanceAudit

### Controls

- Audit Trail: AuditTrail
- Maintenance Date: MaintenanceDate
- ID\_Audit(Hidden): ID\_Auto

## DD. subFormPMFilteredSites

### Record Source

- qryViewPMFilters

### Controls

- COCOM: COCOM
- Site ID: ID\_Site
- Service Branch: ServiceBranch
- Site Name: SiteName
- Site Count: =Count([ID\_Site])
- subFormSiteSurveyAudit

### Record Source

- qrySiteSurveyAudit

### Controls

- Start Date: ActualStartDate
- Audit Trail: AuditTrail
- Finish Date: ActualFinishDate
- ID\_Auto(hidden): ID\_Auto
- Type: Type

## EE. subFormSiteToContactInformation

### Record Source

- qrySiteToContactsList

### Dependencies

- Drop-Down Lists
  - Contact: qryContactsList

### Events

- New Button Click

```
Private Sub btnNew_Click()
On Error GoTo Err_btnNew_Click
    Dim stDocName As String
    Dim stLinkCriteria As String
    stDocName = "ContactInformationByContact"
    DoCmd.OpenForm stDocName, acNormal, , , acFormAdd
Exit_btnNew_Click:
    Exit Sub
Err_btnNew_Click:
    MsgBox Err.Description
    Resume Exit_btnNew_Click
End Sub
```

- Edit Button Click

```
Private Sub Edit_Click()
On Error GoTo Err_Edit_Click
```

```

Dim stDocName As String
Dim stLinkCriteria As String
stDocName = "ContactInformationByContact"
stLinkCriteria = "[ID_Contact] = " & Me![ID_Contact]
DoCmd.OpenForm stDocName, , , stLinkCriteria
Exit>Edit_Click:
Exit Sub
Err>Edit_Click:
MsgBox Err.Description
Resume Exit>Edit_Click
End Sub

```

### Controls

- Billet: Billet
- New(Button): btnNew\_Click
- Edit(Button): Edit\_Click
- ID\_Auto(Hidden): ID\_Auto
- Contact: ID\_Contact

## FF. subFormSurveys

### Record Source

- SiteSurvey

### Order By

- ActualStartDate

### Events

- Before Update

```

Private Sub Form_BeforeUpdate(Cancel As Integer)
If Not CurrentUser = "DBIDS Manager" Then Call Audit_Trail(Me)
End Sub

```

### Controls

- Actual Finish Date: ActualFinishDate
- Actual Start Date: ActualStartDate
- Current Status: CurrentStatus
- Delivered Customer: DeliveredCustomer
- Estimated Finish Date: EstimatedFinishDate
- Estimated Start Date: EstimatedStartDate
- ID\_Auto(Hidden): ID\_Auto
- ID\_SiteSurvey(Hidden): ID\_SiteSurvey
- Notes: Notes
- Team: Team
- Report(Unlabeled): Report
- Returned Telos: ReturnedTelos
- Revised Report: RevisedReport
- ROMReceived: ROMReceived
- Initial Review: Initial Review
- Type: Type

## GG. Switchboard

### Record Source

```

SELECT [tblSwitchboardItems].*, tblUnitInfo.[Title] FROM [tblSwitchboardItems],
tblUnitInfo;

```

## Events

- On Open

```
Private Sub Form_Open(Cancel As Integer)
' Minimize the database window, initialize the switchboard and then load window to
allow Administrator booting.
Dim stDocName As String
' Move to the switchboard page that is marked as the default.
Me.Filter = "[ItemNumber] = 0 AND [Argument] = 'Default' "
Me.FilterOn = True

DoCmd.OpenForm "frmHiddenManager", , , , , acHidden
End Sub
```

- On Current

```
Private Sub Form_Current()
' Update the caption and fill in the list of options.
Me.Caption = Nz(Me![ItemText], "")
FillOptions
End Sub
```

- Private Sub FillOptions

```
Private Sub FillOptions()
' Fill in the options for this switchboard page.
' The number of buttons on the form.
Const conNumButtons = 12

Dim con As Object
Dim rs As Object
Dim stSql As String
Dim intOption As Integer
Dim administrator As Boolean

' If the Current user is in the Administrator's Group Set flag to show
Administrator only switch
board options
administrator = CurrentUserInGroup("Full Permissions")

' Set the focus to the first button on the form,
' and then hide all of the buttons on the form
' but the first. You can't hide the field with the focus.
Me![Option1].SetFocus
For intOption = 2 To conNumButtons
    Me("Option" & intOption).Visible = False
    Me("OptionLabel" & intOption).Visible = False
Next intOption

' Open the table of Switchboard Items, and find
' the first item for this Switchboard Page.
Set con = Application.CurrentProject.Connection
stSql = "SELECT * FROM [tblSwitchboardItems]"
stSql = stSql & " WHERE [ItemNumber] > 0 AND [SwitchboardID] = " &
Me![SwitchboardID]
stSql = stSql & " ORDER BY [ItemNumber];"
Set rs = CreateObject("ADoDB.Recordset")
rs.Open stSql, con, 1 ' 1 = adOpenKeyset

' If there are no options for this Switchboard Page,
' display a message. Otherwise, fill the page with the items.
If (rs.EOF) Then
    Me![OptionLabel1].Caption = "There are no items for this switchboard page"
Else
    While (Not (rs.EOF))
        If rs![AdministratorOnly] And administrator Then
            Me("Option" & rs![ItemNumber]).Visible = True
            Me("OptionLabel" & rs![ItemNumber]).Visible = True
            Me("OptionLabel" & rs![ItemNumber]).Caption = rs![ItemText]
```

```

        ElseIf Not rs![AdministratorOnly] Then
            Me("Option" & rs![ItemNumber]).Visible = True
            Me("OptionLabel" & rs![ItemNumber]).Visible = True
            Me("OptionLabel" & rs![ItemNumber]).Caption = rs![ItemText]
        End If
        rs.MoveNext
    Wend
End If
' Close the recordset and the database.
rs.Close
Set rs = Nothing
Set con = Nothing
End Sub

```

- Private Function HandelButtonClick

```

Private Function HandleButtonClick(intBtn As Integer)
' This function is called when a button is clicked.
' intBtn indicates which button was clicked.
' Constants for the commands that can be executed.
Const conCmdGotoSwitchboard = 1
Const conCmdOpenFormAdd = 2
Const conCmdOpenFormBrowse = 3
Const conCmdOpenReport = 4
Const conCmdCustomizeSwitchboard = 5
Const conCmdExitApplication = 6
Const conCmdRunMacro = 7
Const conCmdRunCode = 8
Const conCmdOpenPage = 9
Const conCmdOpenFormDatasheet = 10
Const conCmdOpenReportForm = 11
Const conCmdToggleBypass = 12
' An error that is special cased.
Const conErrDoCmdCancelled = 2501
Dim con As Object
Dim rs As Object
Dim stSql As String
On Error GoTo HandleButtonClick_Err
' Find the item in the Switchboard Items table
' that corresponds to the button that was clicked.
Set con = Application.CurrentProject.Connection
Set rs = CreateObject("ADoDB.Recordset")
stSql = "SELECT * FROM [tblSwitchboardItems]"
stSql = stSql & "WHERE [SwitchboardID] = " & Me![SwitchboardID] & " AND "
[ItemNumber] = "
& intBtn
rs.Open stSql, con, 1 ' 1 = adOpenKeyset

' If no item matches, report the error and exit the function.
If (rs.EOF) Then
    MsgBox "There was an error reading the Switchboard Items table."
    rs.Close
    Set rs = Nothing
    Set con = Nothing
    Exit Function
End If
Select Case rs![Command]

' Go to another switchboard.
Case conCmdGotoSwitchboard
    Me.Filter = "[ItemNumber] = 0 AND [SwitchboardID] = " & rs![Argument]

' Open a form in Add mode.
Case conCmdOpenFormAdd
    DoCmd.OpenForm rs![Argument], , , , acAdd
' Open a form.
Case conCmdOpenFormBrowse
    DoCmd.OpenForm rs![Argument]
' Open a report.
Case conCmdOpenReport

```

```

        DoCmd.OpenReport rs![Argument], acPreview
        ' Customize the Switchboard.
        Case conCmdCustomizeSwitchboard
            ' Handle the case where the Switchboard Manager
            ' is not installed (e.g. Minimal Install).
            On Error Resume Next
            Application.Run "ACWZMAIN.sbm_Entry"
            If (Err <> 0) Then MsgBox "Command not available."
            On Error GoTo 0
            ' Update the form.
            Me.Filter = "[ItemNumber] = 0 AND [Argument] = 'Default' "
            Me.Caption = Nz(Me![ItemText], "")
            FillOptions
            ' Exit the application.
        Case conCmdExitApplication
            Application.Quit
            ' Run a macro.
        Case conCmdRunMacro
            DoCmd.RunMacro rs![Argument]
            ' Run code.
        Case conCmdRunCode
            Application.Run rs![Argument]
            ' Open a Data Access Page
        Case conCmdOpenPage
            DoCmd.OpenDataAccessPage rs![Argument]
            ' Open a Form in Datasheet Mode
        Case conCmdOpenFormDatasheet
            DoCmd.OpenForm rs![Argument], acFormDS
            ' Open Report Filter Form
        Case conCmdOpenReportForm
            DoCmd.OpenForm "frmOpenReports", , , , , rs![Argument]

            Case conCmdToggleBypass
                CurrentDb.Properties("AllowBypassKey") = Not
                CurrentDb.Properties("AllowBypassKey").Value
                MsgBox "Bypass set to: " &
                CurrentDb.Properties("AllowBypassKey").Value

                ' Any other command is unrecognized.
            Case Else
                MsgBox "Unknown option."
        End Select
        ' Close the recordset and the database.
        rs.Close
    HandleButtonClick_Exit:
    On Error Resume Next
        Set rs = Nothing
        Set con = Nothing
        Exit Function
    HandleButtonClick_Err:
        ' If the action was cancelled by the user for
        ' some reason, don't display an error message.
        ' Instead, resume on the next line.
        If (Err = conErrDocmdCancelled) Then
            Resume Next
        Else
            MsgBox "There was an error executing the command.", vbCritical
            Resume HandleButtonClick_Exit
        End If
    End Function

```

## Controls

- Option1(Button): =HandleButtonClick(1)
- Option2(Button): =HandleButtonClick(2)
- Option3(Button): =HandleButtonClick(3)
- Option4(Button): =HandleButtonClick(4)

- Option5(Button): =HandleButtonClick(5)
- Option6(Button): =HandleButtonClick(6)
- Option7(Button): =HandleButtonClick(7)
- Option8(Button): =HandleButtonClick(8)
- Option9(Button): =HandleButtonClick(9)
- Option10(Button): =HandleButtonClick(10)
- Option11(Button): =HandleButtonClick(11)
- Option12(Button): =HandleButtonClick(12)

## HH. ViewPMFilters

### Record Source

- tbIPMFilters

### Dependencies

- Subforms
  - Filtered View: subFormPMFilteredSites

### Events

- Apply Filter Button Click

```
Private Sub btnApplyFilter_Click()
  On Error GoTo Err_btnApplyFilter_Click
  Me.FilteredView.Form.Filter = IIf (Me.COCOM_Filtered_ And
  Me.Service_Branch_Filtered_, Me.COCOM_Filter & " AND " &
  Me.Service_Branch_Filter, IIf (Me.COCOM_Filtered_, Me.COCOM_Filter, ""))
  Me.FilteredView.Form.FilterOn = True
Exit_btnApplyFilter_Click:
  Exit Sub
Err_btnApplyFilter_Click:
  Beep
  MsgBox "The filter you are trying to apply is formatted incorrectly!"
  Me.FilteredView.Form.Filter = ""
  Me.FilteredView.Form.FilterOn = True
  Resume Exit_btnApplyFilter_Click
End Sub
```

### Controls

- COCOM Filter: COCOMFilter
- COCOM Filtered?(Unlabeled): COCOMFiltered
- Current User: User
- Service Branch Filter: Service Branch Filter
- Service Branch Filtered? (Unlabeled): ServiceBranchFiltered

## **VIII. REPORTS**

The reports in the database are listed here. The following information is listed for each report:

- Record Source – The query or table the report is using to pull its data from.
- Sort – Identifies the fields and order the report is sorted on.
- Group – Identifies the fields that information is grouped by.
- Events – Identifies the events that will generate an automated response and shows the VB Code associated to that event.
- Controls – Shows the data listed on the report and the associated field from the record source from which data is being retrieved.

### **A. rptAuditTrail**

#### **Record Source**

- qryAuditTrailView

#### **Sort**

- COCOM
- Location
- SiteName

#### **Controls**

- COCOM: COCOM
- Location: Location
- Site ID: ID\_Site
- Site Name: SiteName
- Command: CommandName
- Funding(Subreport): subRptAuditTrailFunding
- Maintenance(Subreport): subRptAuditTrailMaintenance
- Site Survey(Subreport): subRptAuditTrailSiteSurvey
- ID\_Auto(Hidden): ID\_Auto
- Date(Unlabeled): = Date()
- Time(Unlabeled): = Time()
- Page#(Unlabeled): ="Page " & [Page] & " of " & [Pages]

### **B. rptContactInformationByContact**

#### **Record Source**

- qryContactsReportByContact

#### **Sort**

- ContactName
- Billet
- Site

#### **Grouping**

- ID\_Contact

## Controls

- Contact Name: ContactName
- Email: Email
- Comm Phone #: CommPhone1
- DSN #: DSN1
- Billet(Unlabeled): Billet
- Site (Unlabeled): Site
- COCOM(Unlabeled): COCOM
- ServiceBranch(Unlabeled): ServiceBranch
- Date(Unlabeled): = Date()
- Time(Unlabeled): = Time()
- Page#(Unlabeled): ="Page " & [Page] & " of " & [Pages]

## C. rptContactInformationBySite

### Record Source

- qryContactsReportBySite

### Sort

- ID\_COCOM
- Location
- ServiceBranch
- SiteName
- Billet
- ContactName

### Grouping

- ID\_COCOM
- ID\_Location
- ID\_ServiceBranch
- ID\_Auto

### Controls

- Billet: Billet
- Contact Name: ContactName
- Email: Email
- Comm Phone #: CommPhone1
- DSN #: DSN1
- COCOM(Unlabeled): COCOM
- Location(Unlabeled): Location
- ServiceBranch(Unlabeled): ServiceBranch
- ID\_Site(Unlabeled): ID\_Site
- SiteName(Unlabeled): SiteName
- Date(Unlabeled): = Date()
- Time(Unlabeled): = Time()
- Page#(Unlabeled): ="Page " & [Page] & " of " & [Pages]

## D. rptMaint

### Record Source

- qryMaintReport

**Sort**

- SiteName

**Controls**

- Site ID: ID\_Site
- Site Name: SiteName
- Maint: Maint
- Maint Start Date: MaintStartDate
- Date(Unlabeled): =Date()
- Time(Unlabeled): =Time()
- Page#(Unlabeled): ="Page " & [Page] & " of " & [Pages]

**E. rptPendingSite****Record Source**

- qryPendingSiteReport

**Sort**

- ID\_COCOM
- Location\_State
- Site

**Controls**

- Site: Site
- Site Survey(Subreport): subRptSiteSurvey
- Funding(Subreport): subRptFunding
- ID\_Auto(Hidden): ID\_Auto
- Date(Unlabeled): =Date()
- Time(Unlabeled): =Time()
- Page#(Unlabeled): ="Page " & [Page] & " of " & [Pages]

**F. rptQuickStats****Record Source**

- qryQuickStatsReport

**Sort**

- ID\_COCOM (inv)
- Location\_State
- ID\_ServiceBranch

**Grouping**

- ID\_COCOM
- ID\_Location
- ID\_ServiceBranch

**Controls**

- Branch(In ID\_COCOM grouping): COCOM
- Branch(In ID\_Location grouping): Location\_State
- Branch(In ID\_ServiceBranch grouping): ServiceBranch
- In all Group Headings and Report Header:
- Total Planned: =Sum(TotalPlanned)

- Total Installed: =Sum(TotalInstalled)
- Primary Planned: =Sum(PrimaryPlanned)
- Primary Installed: =Sum(PrimaryInstalled)
- Secondary Planned: =Sum(SecondaryPlanned)
- Secondary Installed: =Sum(SecondaryInstalled)
- Registration Planned: =Sum(RegistrationPlanned)
- Registration Installed: =Sum(RegistrationInstalled)
- LEO Planned: =Sum(LEOPlanned)
- LEO Installed: =Sum(LEOInstalled)
- Gates Planned: =Sum(GatesPlanned)
- Gates Installed: =Sum(GatesInstalled)
- MP Eff Gates Planned: =Sum(ManPowerEfficientGatePlanned)
- MP Eff Gates Installed: =Sum(ManPowerEfficientGateInstalled)
- Mobile Kits Planned: =Sum(MobileKitsPlanned)
- Mobile Kits Installed: =Sum(MobileKitsInstalled)
- Visitor Center Planned: =Sum(VisitorCenterPlanned)
- Visitor Center Installed: =Sum(VisitorCenterInstalled)
- Extra Handhelds Planned: =Sum(ExtraHandheldsPlanned)
- Extra Handhelds Installed: =Sum(ExtraHandheldsInstalled)
- Registrants Total: =Sum(TotalArchivedCurrent)
- Registrants Current: =Sum(TotalCurrent)
- Registrants Cards: =Sum(DBIDSCards)
- Date(Unlabeled): =Date()
- Time(Unlabeled): =Time()
- Page#(Unlabeled): ="Page " & [Page] & " of " & [Pages]

## G. rptQuickStatsAudit

### Record Source

- qryQuickStatsReport

### Sort

- ID\_COCOM (inv)
- LocationState
- ID\_ServiceBranch
- SiteName

### Grouping

- ID\_COCOM
- ID\_Location
- ID\_ServiceBranch

### Controls

- Branch(In ID\_COCOM grouping): COCOM
- Branch(In ID\_Location grouping): Location\_State
- Branch(In ID\_ServiceBranch grouping): ServiceBranch
- In all Group Headings and Report Header:
- Total Planned: =Sum(TotalPlanned)
- Total Installed: =Sum(TotalInstalled)
- Primary Planned: =Sum(PrimaryPlanned)
- Primary Installed: =Sum(PrimaryInstalled)
- Secondary Planned: =Sum(SecondaryPlanned)

- Secondary Installed: =Sum(SecondaryInstalled)
- Registration Planned: =Sum(RegistrationPlanned)
- Registration Installed: =Sum(RegistrationInstalled)
- LEO Planned: =Sum(LEOPlanned)
- LEO Installed: =Sum(LEOInstalled)
- Gates Planned: =Sum(GatesPlanned)
- Gates Installed: =Sum(GatesInstalled)
- MP Eff Gates Planned: =Sum(ManPowerEffecientGatePlanned)
- MP Eff Gates Installed: =Sum(ManPowerEffecientGateInstalled)
- Mobile Kits Planned: =Sum(MobileKitsPlanned)
- Mobile Kits Installed: =Sum(MobileKitsInstalled)
- Visitor Center Planned: =Sum(VisitorCenterPlanned)
- Visitor Center Installed: =Sum(VisitorCenterInstalled)
- Extra Handhelds Planned: =Sum(ExtraHandheldsPlanned)
- Extra Handhelds Installed: =Sum(ExtraHandheldsInstalled)
- Registrants Total: =Sum(TotalArchivedCurrent)
- Registrants Current: =Sum(TotalCurrent)
- Registrants Cards: =Sum(DBIDSCards)
- Total Planned: TotalPlanned
- Total Installed: TotalInstalled
- Primary Planned: PrimaryPlanned
- Primary Installed: PrimaryInstalled
- Secondary Planned: SecondaryPlanned
- Secondary Installed: SecondaryInstalled
- Registration Planned: RegistrationPlanned
- Registration Installed: RegistrationInstalled
- LEO Planned: LEOPlanned
- LEO Installed: LEOInstalled
- Gates Planned: GatesPlanned
- Gates Installed: GatesInstalled
- MP Eff Gates Planned: ManPowerEffecientGatePlanned
- MP Eff Gates Installed: ManPowerEffecientGateInstalled
- Mobile Kits Planned: MobileKitsPlanned
- Mobile Kits Installed: MobileKitsInstalled
- Visitor Center Planned: VisitorCenterPlanned
- Visitor Center Installed: VisitorCenterInstalled
- Extra Handhelds Planned: ExtraHandheldsPlanned
- Extra Handhelds Installed: ExtraHandheldsInstalled
- Registrants Total: TotalArchivedCurrent
- Registrants Current: TotalCurrent
- Registrants Cards: DBIDSCards
- Date(Unlabeled): =Date()
- Time(Unlabeled): =Time()
- Page#(Unlabeled): ="Page " & [Page] & " of " & [Pages]

## H. rptQuickStatsCOCOM

### Record Source

- qryQuickStatsReport
-

### **Sort**

- ID\_COCOM (inv)
- ServiceBranch

### **Grouping**

- ID\_COCOM
- ID\_ServiceBranch

### **Controls**

- Branch(In ID\_COCOM grouping): COCOM
- Branch(In ID\_ServiceBranch grouping): ServiceBranch
- In all Group Headings and Report Header:
- Total Planned: =Sum(TotalPlanned)
- Total Installed: =Sum(TotalInstalled)
- Primary Planned: =Sum(PrimaryPlanned)
- Primary Installed: =Sum(PrimaryInstalled)
- Secondary Planned: =Sum(SecondaryPlanned)
- Secondary Installed: =Sum(SecondaryInstalled)
- Registration Planned: =Sum(RegistrationPlanned)
- Registration Installed: =Sum(RegistrationInstalled)
- LEO Planned: =Sum(LEOPlanned)
- LEO Installed: =Sum(LEOInstalled)
- Gates Planned: =Sum(GatesPlanned)
- Gates Installed: =Sum(GatesInstalled)
- MP Eff Gates Planned: =Sum(ManPowerEffecientGatePlanned)
- MP Eff Gates Installed: =Sum(ManPowerEffecientGateInstalled)
- Mobile Kits Planned: =Sum(MobileKitsPlanned)
- Mobile Kits Installed: =Sum(MobileKitsInstalled)
- Visitor Center Planned: =Sum(VisitorCenterPlanned)
- Visitor Center Installed: =Sum(VisitorCenterInstalled)
- Extra Handhelds Planned: =Sum(ExtraHandheldsPlanned)
- Extra Handhelds Installed: =Sum(ExtraHandheldsInstalled)
- Registrants Total: =Sum(TotalArchivedCurrent)
- Registrants Current: =Sum(TotalCurrent)
- Registrants Cards: =Sum(DBIDSCards)
- Date(Unlabeled): =Date()
- Time(Unlabeled): =Time()
- Page#(Unlabeled): ="Page " & [Page] & " of " & [Pages]

## **I. rptSiteSurvey**

### **Record Source**

- qrySiteSurveyReport

### **Sort**

- SiteName
- ActualStartDate

### **Grouping**

- ID\_Site

## Events

- On No Data:

```
Private Sub Report_NoData(Cancel As Integer)
    MsgBox "There are no records meeting that criteria!@"
    Cancel = True
End Sub
```

## Controls

- Actual Duration: ActualDuration
- Actual Finish: ActualFinishDate
- Actual Start: ActualStartDate
- COCOM: COCOM
- Current Status: CurrentStatus
- Delivered Customer: DeliveredCustomer
- Est Duration: EstDuration
- Est Finish: EstimatedFinishDate
- Est Start: EstimatedStartDate
- Initial Review: InitialReview
- Notes: Notes
- Returned Telos: ReturnedTelos
- Revised Report: RevisedReport
- ROM Received: ROMReceived
- Service Branch: ServiceBranch
- Site ID: ID\_Site
- Site Name: SiteName
- Team: Team
- Date(Unlabeled): =Date()
- Time(Unlabeled): =Time()
- Page#(Unlabeled): ="Page " & [Page] & " of " & [Pages]

## J. subRptAuditTrailFunding

### Record Source

- qryFundingAudit

### Sort

- FundingDate

### Controls

- Date: FundingDate
- Audit Trail: AuditTrail
- ID\_Auto(Hidden): ID\_Auto

## K. subRptAuditTrailMaintenance

### Record Source

- qryMaintenanceAudit

### Sort

- MaintenanceDate

**Controls**

- Date: MaintenanceDate
- Audit Trail: AuditTrail
- ID\_Auto(Hidden): ID\_Auto

**L. subRptAuditTrailSiteSurvey****Record Source**

- qrySiteSurveyAudit

**Sort**

- ActualFinishDate

**Controls**

- Start Date: ActualStartDate
- Finish Date: ActualFinishDate
- Type: Type
- Audit Trail: AuditTrail
- ID\_Auto(Hidden): ID\_Auto

**M. subRptFunding****Record Source**

- Funding

**Sort**

- FundingDate

**Controls**

- Date: FundingDate
- Notes: Notes
- Reason: Reason
- Received: Received
- Required: Required
- Source: Source
- Received Column Total(Unlabeled): =Sum([Received])
- Required Column Total(Unlabeled): =Sum([Required])
- ID\_Auto(Hidden): ID\_Auto

**N. subRptSiteSurvey****Record Source**

- SiteSurvey

**Sort**

- ActualFinishDate

**Controls**

- Date: ActualFinishDate
- Duration: =[ActualFinishDate]-[ActualStartDate]+1
- Notes: Notes
- Report Date: RevisedReport
- Team: Team

THIS PAGE INTENTIONALLY LEFT BLANK

## APPENDIX J—VISUAL BASIC MODULE

It was necessary to create a code module for a few of the VB Functions. This is where sections of code were placed, that would be utilized repetitively by multiple forms.

### A. Audit Trail Delete

```
Public Function Audit_Trail_Delete(frm As Form)

On Error GoTo Err_Audit_Trail_Delete

'Based on:
'ACC2000: How to Create an Audit Trail of Record Changes in a Form
'http://support.microsoft.com/default.aspx?scid=kb;en-us;197592

Dim ctl As Control
Dim sUser As String
sUser = CurrentUser

'Check each data entry control for change and record old value of the control.

frm!AuditTrail = "Record deleted on: " & Now & " by " & sUser & " " &
frm!AuditTrail.Value
frm.Requery

Exit_Audit_Trail_Delete:
Exit Function

Err_Audit_Trail_Delete:
If Err.Number = 64353 Then 'Operation is not supported for this type of object.
    Exit Function
ElseIf Err.Number = 2475 Then 'You entered an expression that requires a form to
be the active window
    Beep
    MsgBox "A form is required to be the active window!", vbCritical, "Invalid
Active Window"
Else
    'Beep
    'MsgBox Err.Number & " - " & Err.Description
End If
Resume Exit_Audit_Trail_Delete

End Function
```

### B. Audit Trail

```
Public Function Audit_Trail(frm As Form)

On Error GoTo Err_Audit_Trail

'ACC2000: How to Create an Audit Trail of Record Changes in a Form
'http://support.microsoft.com/default.aspx?scid=kb;en-us;197592

Dim ctl As Control
Dim sUser As String
sUser = CurrentUser

'If new record, record it in audit trail and exit function.
If frm.NewRecord = True Then
    frm!AuditTrail = frm!AuditTrail & "New Record added on " & Now & " by " &
sUser & ";" & vbCrLf
```

```

        Exit Function
End If

'Check each data entry control for change and record old value of the control.
For Each ctl In frm.Controls

'Only check data entry type controls.
Select Case ctl.ControlType
    Case acTextBox, acComboBox, acListBox, acOptionGroup, acCheckBox
        If ctl.name = "AuditTrail" Then GoTo TryNextControl 'Skip AuditTrail
field.
        'If new and old value do not equal
        If ctl.Value <> ctl.OldValue Or Len(ctl.Value) <> Len(ctl.OldValue) Then
            frm!AuditTrail = frm!AuditTrail & ctl.name & ": Changed From: " &
ctl.OldValue & ", To: " & ctl.Value & " by " & sUser & "; " & Now & vbCrLf

        'If old value is Null and new value is not Null
        ElseIf IsNull(ctl.OldValue) And Len(ctl.Value) > 0 Or ctl.OldValue = ""
And Len(ctl.Value) > 0 Then
            frm!AuditTrail = frm!AuditTrail & ctl.name & ": Previously Null,
New Value: " & ctl.Value & " by " & sUser & "; " & Now & vbCrLf
            'If new value is Null and old value is not Null
            ElseIf IsNull(ctl.Value) And Len(ctl.OldValue) > 0 Or ctl.Value = "" And
Len(ctl.OldValue) > 0 Then
                frm!AuditTrail = frm!AuditTrail & ctl.name & ": Changed From: " &
ctl.OldValue & ", To: Null" & " by " & sUser & "; " & Now & vbCrLf
            End If
        End Select

TryNextControl:
    Next ctl

Exit_Audit_Trail:
    Exit Function

Err_Audit_Trail:
    If Err.Number = 64535 Then 'Operation is not supported for this type of object.
        Exit Function
    ElseIf Err.Number = 2475 Then 'You entered an expression that requires a form to
be the active window
        Beep
        MsgBox "A form is required to be the active window!", vbCritical, "Invalid
Active Window"
    Else
        'Beep
        'MsgBox Err.Number & " - " & Err.Description
    End If
    Resume Exit_Audit_Trail

End Function

```

## C. Get Current User Filter

```

Public Function GetCurrentUserFilter(FullClause As Boolean, COCOMOnly As Boolean) As
String
On Error GoTo Err_GetCurrentUserFilter
    Dim con As Object
    Dim rs As Object
    Dim stSql As String
    Dim strFilter As String
    strFilter = ""

    Set con = Application.CurrentProject.Connection
    stSql = "SELECT * FROM [qryPMFilter] WHERE [User] = '" & CurrentUser & "'"
    Set rs = CreateObject("ADODB.Recordset")
    rs.Open stSql, con, 1
    If (rs.EOF) Then
        GetCurrentUserFilter = ""
    Else
        If (COCOMOnly) Then

```

```

        strFilter = rs![COCOMOnly]
    Else
        strFilter = rs![PMFilter]
    End If
    If (FullClause) And (strFilter <> "") Then
        strFilter = "WHERE " & strFilter
    ElseIf (strFilter <> "") Then
        strFilter = " AND " & strFilter
    End If
    GetCurrentUserFilter = strFilter
End If

rs.Close
Set rs = Nothing
Set con = Nothing

Exit_GetCurrentUserFilter:
    Exit Function

Err_GetCurrentUserFilter:
    MsgBox "There was an error gathering the PM Filter."
    MsgBox Err.Description
    GetCurrentUserFilter = ""
    Resume Exit_GetCurrentUserFilter
End Function

```

## D. Current User In Group

```

Public Function CurrentUserInGroup(GroupName As String) As Boolean
'*****
' Purpose: Determines if the current user belongs to the specified
'           group.
' Accepts: The name of a group.
' Returns: True if the current user is a member of the specified
'           group, False if the current user is not a member of
'           the group.
' Assumes: The existence of a user called Developer in the Admins
'           group, with no password.
' Source: http://support.microsoft.com/?kbid=210331
'*****

On Error GoTo err_CurrentUserInGroup

Dim MyWorkSpace As Workspace, i As Integer
Dim MyGroup As Group, MyUser As User

' Create a new workspace as a member of the Admins group.
Set MyWorkSpace = DBEngine.CreateWorkspace("SPECIAL", "Developer", "")

Set MyGroup = MyWorkSpace.Groups(GroupName)
Set MyUser = MyWorkSpace.Users(CurrentUser())
For i = 0 To MyGroup.Users.count - 1
    If MyGroup.Users(i).name = MyUser.name Then
        CurrentUserInGroup = True
        Exit Function
    End If
Next i

CurrentUserInGroup = False
MyWorkSpace.Close
Exit Function

err_CurrentUserInGroup:
    If Err = 3265 Then
        MsgBox UCase(GroupName) & " isn't a valid group name", 16, "Error"
        CurrentUserInGroup = False
    ElseIf Err = 3029 Then
        MsgBox "The account used to create the workspace does not exist"
    Else: MsgBox Error(Err)
    End If

```

```
MyWorkSpace.Close  
Exit Function  
  
End Function
```

## INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center  
Ft. Belvoir, Virginia
2. Dudley Knox Library  
Naval Postgraduate School  
Monterey, California
3. Marine Corps Representative  
Naval Postgraduate School  
Monterey, California
4. Director, Training and Education, MCCDC, Code C46  
Quantico, Virginia
5. Director, Marine Corps Research Center, MCCDC, Code C40RC  
Quantico, Virginia
6. Marine Corps Tactical Systems Support Activity (Attn: Operations Officer)  
Camp Pendleton, California
7. LtCol Karl Pfeifer  
Naval Postgraduate School  
Monterey, California
8. Cori Mazik (DBIDS Operations Chief)  
Defense Manpower Data Center (DMDC)  
Fort Ord, California
9. Dan Boger  
Naval Postgraduate School  
Monterey, California